



HAL
open science

Formalisation des contraintes pragmatiques pour la génération des énoncés en dialogue homme-machine multi-locuteurs

Vladimir Popescu

► **To cite this version:**

Vladimir Popescu. Formalisation des contraintes pragmatiques pour la génération des énoncés en dialogue homme-machine multi-locuteurs. Interface homme-machine [cs.HC]. Institut National Polytechnique de Grenoble - INPG, 2008. Français. NNT: . tel-00343846

HAL Id: tel-00343846

<https://theses.hal.science/tel-00343846>

Submitted on 2 Dec 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT POLYTECHNIQUE DE GRENOBLE

N^o. attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--	--

THÈSE EN COTUTELLE INTERNATIONALE

pour obtenir le grade de
DOCTEUR DE L'Institut Polytechnique de Grenoble
et
de l'Université « Politehnica » de Bucarest
Spécialité : Mathématiques et Informatique

préparée au Laboratoire d'Informatique de Grenoble
dans le cadre de l'École Doctorale « **Mathématiques, Sciences et Technologies de
l'Information, Informatique** »
et à la Faculté d'Electronique, Télécommunications et Technologie de l'Information de
Bucarest

présentée et soutenue publiquement par

Vladimir POPESCU

le 28 novembre 2008

**Formalisation des contraintes pragmatiques pour la
génération des énoncés en dialogue homme-machine
multi-locuteurs**

DIRECTEUR DE THÈSE : Jean CAELEN

CO-DIRECTEUR DE THÈSE : Corneliu BURILEANU

JURY

M. Denis VERNANT, Professeur, Université Pierre Mendès France, Grenoble
M. Nicholas ASHER, Directeur de Recherches, IRIT-CNRS, Toulouse
M. Dan TUFIȘ, Directeur de Recherches, RACAI-Académie Roumaine, Bucarest
M. Horia-Nicolai TEODORESCU, Professeur, Université « Gheorghe Asachi », Iași
M. Jean CAELEN, Directeur de Recherches, LIG-CNRS, Grenoble
M. Corneliu BURILEANU, Professeur, Université « Politehnica », Bucarest

Président
Rapporteur
Rapporteur
Examineur
Directeur de thèse
Co-directeur de thèse

Remerciements

Nous n’aurions jamais réussi à mener à bien cette thèse sans le support constant de nos deux directeurs de thèse, Jean Caelen et Corneliu Burileanu. Nous les remercions donc pour tout leur soutien le long de ces trois dernières années pendant tout le déroulement de cette thèse.

Un grand remerciement à Jean Caelen, pour nous avoir fait constamment confiance, en acceptant, toutefois de manière critique, nos idées, en essayant de les améliorer, cela tout en nous faisant attentif aux pistes moins fructueuses, auxquelles nous nous serions engagé.

Un grand remerciement à Corneliu Burileanu également, tout d’abord pour nous avoir toujours fait confiance et avoir cru (parfois même plus que nous même) dans nos capacités à mener à bien un travail de recherche, ensuite pour nous avoir encouragé à effectuer cette thèse en France.

Nous remercions également nos rapporteurs, Nicholas Asher et Dan Tufiş, pour avoir accepté de faire partie de notre jury, ainsi que pour leur lecture attentive du manuscrit, et pour les suggestions pertinentes qu’ils nous ont faites pour l’améliorer.

Nous remercions aussi les autres membres du jury, Horia-Nicolai Teodorescu et Denis Vernant, pour avoir accepté de porter leur attention sur notre travail, en nous faisant des remarques pertinentes, qui nous aideront certainement à améliorer notre perspective sur les recherches abordées.

Un grand remerciement aussi à Jean-François Serignat, pour nous avoir accueilli au sein de l’équipe GETALP (au début, GEOD) au Laboratoire d’Informatique de Grenoble.

Nous remercions également nos collègues à Grenoble – notamment (dans l’ordre alphabétique) Dima Achlawafa-Mufti, Gwenn Boussard, Stéphane Chaillol, Patricia Dominguez-Sanchez, Achille Falaise, Viet-Bac Le, Nadine Mandran, Brigitte Meillon, Anas Ouayouch, Vu-Minh Quang, Sopheap Seng et Michel Vacher, pour avoir su rendre très agréable l’ambiance au laboratoire, et aussi pour les moments conviviaux que nous avons partagés ensemble.

En qui concerne l’équipe qui nous a accueilli à Bucarest, lors de nos différents séjours en 2006 et en 2007, nous remercions tout particulièrement Dragoş Burileanu, surtout pour les moments très nombreux que nous avons partagés pendant le travail, mais aussi pendant les pauses de café, dans les soirées tardives à l’Université « Politehnica » ; nous le remercions également pour ses conseils toujours pertinents concernant divers aspects dans la carrière de chercheur.

Nous remercions aussi tout particulièrement Inge Gavăt et Cristian Negrescu, pour avoir accepté de suivre nos travaux, lors de nos séjours à Bucarest, ainsi que pour leurs conseils toujours bien intentionnés.

Nous remercions également nos collègues à Bucarest – notamment (dans l’ordre alphabétique) Andi Buzo, Amelia Ciobanu, Aurelian Derviş, Diana Ghelmez (Haneş), Nicolae Militaru,

Doru Munteanu, Cristina Petrea et Mihnea Udrea, pour avoir toujours été là où nous avons besoin d'eux, et aussi pour les moments agréables que nous avons passés ensemble lors de ces séjours.

Enfin, cette thèse n'aurait pas pu être menée à bien sans le soutien constant des amis, aussi en France qu'en Roumanie ou ailleurs. Ainsi, pour nos amis français, nous remercions infiniment Christine Plumejeaud, pour avoir été à nos côtés dans des moments critiques dans notre vie, et aussi pour avoir été une personne avec qui nous avons partagé beaucoup plus que des moments simplement conviviaux.

Nous remercions aussi Carole Adam, Virginie Demeure, Anne Garcia-Fernandez, Benoît Gaudou et Linda Onu, surtout pour les moments bien agréables que nous avons partagés au delà des conférences où nous nous étions rencontrés. Ensuite, nous remercions Tanel Alumaë, Timo Bauman, Luciana Benotti, Mathilde Dargnat, Alexandre Denis, Marco Dinarelli, Jacques Jayez, Manuel Kirschner, Natalia Klyueva-Perkukova, Christina Kuehnel, Frédéric Landragin, Georges Linarès, Dimitrios Lyras, Laurent Mazuel, Ana Mendes, Francesco Nesta, Sylwia Ozdowska, Matthieu Quignard, Yves Scherrer, Alexander Schmitt, Marc Staudacher et Petra-Maria Strauß, pour les moments conviviaux que nous avons passés à l'occasion des nombreuses conférences ou écoles d'été où nous avons participé.

Nous remercions également nos amis roumains, notamment Bogdan Ionescu, Dorin Panaitopol et Gabriel Vasile, pour les moments inoubliables que nous avons passés ensemble en France et en Roumanie. Un grand merci également à Anamaria Necşulea, pour des moments plus qu'agréables que nous avons passés ensemble, aussi que pour nos rencontres « littéraires ».

Ensuite, nous remercions nos amis à Bucarest, sans lesquels nos séjours en Roumanie auraient été beaucoup moins qu'ils ne l'ont été : nous pensons notamment à Cătălin Oprea, à Corina Panaitopol et à Sorin Paşa ; nous les remercions pour tout ce qu'ils ont fait pour nous, même par leur simple présence dans notre vie. Ensuite, nous remercions également nos autres amis roumains, qui ont rendu et rendent toujours (si l'occasion apparaît) notre vie quotidienne très agréable, notamment (dans l'ordre alphabétique) Adrian Busuioc, Raluca Ionescu, Radu Lăzărescu, Irina et Sorin Rusu, Daniel Sillion, et aussi d'autres, trop nombreux pour les lister ici.

Une mention particulière aussi pour nos amis qui se trouvent ailleurs, aux Etats Unis ou à Singapour, mais qui ont souvent rendu nos vacances en Roumanie bien agréables – notamment Andrei Aron, Ionuț Constandache, Andrei Hagiescu et Marian Mihăilescu.

Nous remercions également les chercheurs que nous avons rencontrés à des congrès, notamment Laurence Danlos, Markus Egg, Kristiina Jokinen, Wolfgang Minker, Sergei Nirenburg, Marjorie McShane, Catherine Pelachaud, Paul Piwek, David Schlangen et Manfred Stede, pour les discussions souvent inspiratrices que nous avons menées.

Un grand merci va aussi à notre famille, pour nous avoir toujours encouragé et soutenu, lors de toute notre existence.

Résumé

Lors de notre travail de thèse, nous avons développé un cadre formel pour contrôler certains aspects concernant la production des énoncés en dialogue homme-machine à plusieurs locuteurs.

Ainsi, nous contrôlons le processus de génération des énoncés en dialogue à plusieurs locuteurs, en quatre étapes, pour chaque tour de parole de la machine :

(i) La structure de discours qui représente le dialogue est mise à jour ; pour cela, nous proposons une émulation, dans une logique du premier ordre, d'un fragment de la théorie SDRT (« Segmented Discourse Representation Theory »).

(ii) La structure de discours du dialogue et les sémantiques des énoncés que la machine est censée produire sont utilisées pour mettre à jour les *listes d'engagements* des locuteurs ; un engagement représente un fait auquel un locuteur s'est engagé, à travers la parole. A leur tour, ces engagements sont utilisés pour piloter des décisions concernant le réglage du degré de force illocutoire de ces énoncés.

(iii) Les listes d'engagements sont filtrées et placées dans une pile pour chaque locuteur, ce qui représente un traçage sur le processus de mise à jour des connaissances des locuteurs ; ces piles, avec les relations rhétoriques entre les énoncés dialogiques sont ensuite utilisées pour prendre des décisions concernant les ellipses sémantiques autorisées.

(iv) La structure de discours et les listes d'engagements sont utilisées pour le choix lexical des connecteurs concessifs (*mais, quand même, pourtant* et *bien que*) entre les énoncés dans un même tour de parole ; pour réaliser ce *desideratum*, les énoncés susceptibles d'être reliés par ces connecteurs sont ordonnés du point de vue argumentatif.

Dans une perspective de satisfaction des contraintes, chacune des premières trois étapes précédemment énoncées représente un ensemble d'instances de problèmes de résolution avec contraintes ; ces instances sont *concaténées*, au sens où elles sont appliquées de manière séquentielle, avec une priorité plus importante attribuée, de manière implicite, aux étapes exécutées précédemment dans la chaîne des traitements. Néanmoins, la quatrième étape n'est qu'en partie une instance de problème de satisfaction de contraintes : tandis que le choix proprement dit des connecteurs discursifs est à base de contraintes, l'ordonnement argumentatif des énoncés a besoin d'informations *a posteriori*, donc une approche purement à base de contraintes n'est pas applicable telle quelle.

Les contributions principales de cette thèse portent sur :

- la proposition d'un cadre à base de contraintes pour piloter les décisions en génération des énoncés ;
- la conception d'un composant de structuration rhétorique des répliques dialogiques ; pour

ce faire, nous avons pris comme élément de départ la théorie SDRT et les actes de langage qui caractérisent les visées illocutoires des énoncés ; ensuite, nous avons proposé un ensemble de sémantiques dans une logique du premier ordre, pour certaines relations rhétoriques de la SDRT ; ces sémantiques subissent ensuite des contraintes contextuelles, que nous avons formalisées à partir de la notion de grammaire abstraite d'attributs, utilisée dans la vérification des programmes informatiques ;

- le couplage entre l'aspect structurel concernant la représentation rhétorique des énoncés en dialogue, et des phénomènes linguistiques de surface, notamment le contrôle de l'ellipse sémantique, le calcul du degré de force illocutoire, et le choix des connecteurs discursifs ; ainsi, pour les ellipses sémantiques, nous nous avons modélisé les connaissances ou faits auxquels les interlocuteurs se sont engagés, pour ainsi utiliser la structure rhétorique afin de décider sur la pertinence des constructions elliptiques ; pour le degré de force illocutoire, nous avons couplé la structure rhétorique avec les engagements publiquement assumés par les locuteurs, en posant comme critère de succès pour le dialogue l'équivalence sémantique entre les engagements des interlocuteurs ; pour les connecteurs, nous avons proposé un ensemble de contraintes rhétoriques sur leur choix, en passant également par des considérations concernant les valences argumentatives des énoncés ;
- l'extension de toutes ces démarches au dialogue à plusieurs locuteurs ; pour ce faire, nous avons considéré ce type de dialogue comme résultant de certaines superpositions de dialogues à deux locuteurs, en formalisant ensuite les contraintes induites par la simultanéité entre ces dialogues.

Toutes ces contributions ont été validées sur des dialogues typiques, autour d'une tâche de réservation de livres dans une bibliothèque publique.

Abstract

During our thesis work, we have designed a formal framework for controlling certain aspects of utterance production in multi-party human-computer dialogue.

Hence, the pragmatic control of the machine utterance generation process, in multi-party dialogue, is accomplished in four stages, for each speech turn due to be produced:

(i) The discourse structure representing the dialogue is updated.

(ii) The discourse structure of the dialogue and the semantics of the utterance due to be produced are used to update the *commitment stores* of the speakers; in turn, these are used to trigger decisions regarding the adjustment of the degree of strength for the illocutionary force performed by the utterance.

(iii) The commitment stores are filtered, in that semantic predicates are placed in a stack, that constitutes a trace of speakers' knowledge updating process; these stacks, along with rhetorical relations in the discourse structure are then used to make decisions regarding the semantic ellipsis authorized.

(iv) The discourse structure, along with the commitment stores of the speakers, are used in lexically selecting concessive connectors (*but, although, yet, and however* between utterances in the same speech turn; in order to accomplish this, utterances susceptible of being connected *via*

these words are argumentatively ordered.

From a constraint satisfaction perspective, each of the first three stages shown above represent a set of instances of constraint solving problems, that are *concatenated*, in that they are applied in a sequential manner, with higher priority implicitly given to earlier stages. However, the fourth stage is only partially an instance of a constraint satisfaction problem: while the actual selection of connective words is constraint-based, the argumentative ordering of utterances needs posterior information, hence a pure constraint satisfaction approach is less applicable.

Our main contributions concern:

- the proposal of a constraint-based framework for driving utterance generation decisions;
- the design of a component for rhetorically structuring dialogue utterances; for this, we took as a departure point SDRT and the speech acts that characterise the illocutionary goals of the utterances; then, we have proposed a set of semantics in a first-order logic, for certain rhetorical relations in SDRT; these semantics are further contextually constrained; these contextual constraints are formalized using the notion of abstract attribute grammar, used in computer program verification;
- the coupling between structural aspects regarding the rhetorical representation of the utterances, and linguistic phenomena, such as semantic ellipsis control, illocutionary force degree computation, and the lexical choice of some discourse connectives; thus, concerning semantic ellipsis, we have modeled knowledge or facts that the interlocutors are committed to, in order to use then the rhetorical structure for deciding on the relevance of elliptical constructions; for the illocutionary force degree, we have coupled rhetorical structure to interlocutors' public commitments, setting as a dialogue success criterion the semantic equivalence between these commitment stores; for the connectives, we have proposed a set of rhetorical constraints on their choice, also relying on argumentative considerations for computing some of these connectors.
- the extension of all this framework to multi-party human-computer dialogues; in order to do this, we have considered multi-party dialogues as resulting from certain superpositions of dual-party dialogues, then formalizing the constraints induced by the simultaneity between these dialogues.

All these contributions have been validated on typical dialogues concerning a book reservation task in a public library.

Sommaire

Remerciements	i
Résumés	iii
1 Introduction	1
2 Etat de l'art	29
3 Filtrage local des relations rhétoriques	53
4 Filtrage contextuel des relations rhétoriques	75
5 Structuration rhétorique en dialogue multi-locuteurs	99
6 Contrôle de l'ellipse sémantique	125
7 Le degré de force illocutoire	145
8 Génération des connecteurs	173
9 Conclusions	199
A Degré de force illocutoire et réalisation de surface	211
B Glossaire	221
C La SDRT	229

Chapitre 1

Introduction : contraintes pragmatiques en génération pour le dialogue à plusieurs locuteurs

1.1 Introduction

Le sujet de notre thèse porte sur la génération de réponses en dialogue homme-machine. On sait toute l'importance de ce niveau de traitement pour donner à la machine un maximum de naturel dans ses énoncés. En effet ces derniers doivent être non seulement pertinents et compréhensibles mais également cohérents entre eux et adaptés à l'interlocuteur, ce qui veut dire construits dans un terrain commun avec, et pour lui. Il s'agit donc de prendre en compte à la fois le contexte et l'interlocuteur dans la situation où ils se présentent tous les deux pour réaliser une tâche. Ceci est essentiel non seulement pour des raisons ergonomiques (efficacité, précision dans la tâche) mais aussi pour des raisons psychologiques, car la machine reflète par ses énoncés ses capacités « cognitives » – souvent une machine montre des capacités supérieures pour produire des énoncés que pour comprendre ce qui lui donne ainsi une image d'intelligence erronée qui finit par troubler l'utilisateur – qui sont essentielles pour maintenir le dialogue.

Notre thèse met donc l'emphase sur la pragmatique du dialogue dans ses différents aspects, cohérence structurelle, pertinence des énoncés, adaptation à la tâche et à l'utilisateur (niveau de connaissances et relation sociale), adéquation à l'usage linguistique (il serait en effet gênant de ne pas utiliser la concision, les facilités d'expression, les raccourcis, etc., que permet la langue et qui sont en relation directe avec le niveau sémantico-pragmatique).

Nous nous sommes également intéressés au dialogue à plusieurs locuteurs, non seulement pour étendre le problème du dialogue à des cas complexes mais aussi pour mesurer la solidité de notre approche pour deux locuteurs. En ce sens nous prenons un risque pour la modélisation car il n'est pas certain qu'un dialogue homme-machine à plusieurs locuteurs ait des applications pratiques.

Ainsi, en substance, notre thèse porte sur le contrôle du processus de génération des énoncés en dialogue à plusieurs locuteurs, qui se réalise en quatre étapes, pour chaque tour de parole de

la machine :

1. nous calculons la structure du dialogue ;
2. nous utilisons la structure du dialogue et les intentions des énoncés pour piloter des décisions concernant le réglage de la forme de surface de ces énoncés ;
3. nous réalisons un traçage sur le processus de mise à jour des connaissances des locuteurs ; ensuite, nous utilisons ce traçage pour décider ce qui est pertinent d'exprimer sous forme linguistique, dans l'intention communicationnelle ;
4. nous utilisons la structure du dialogue et les connaissances des locuteurs dans le choix lexical des connecteurs concessifs (*mais, quand même, pourtant et bien que*) entre les énoncés dans un même tour de parole.

Dans une perspective de satisfaction des contraintes, chacune de ces étapes précédemment énoncées représente un ensemble d'instances de problèmes de résolution avec contraintes ; ces instances sont *concaténées*, au sens où elles sont appliquées de manière séquentielle, avec une priorité plus importante attribuée, de manière implicite, aux étapes exécutées précédemment dans la chaîne des traitements. Plus loin dans ce chapitre nous décrirons et motiverons ces étapes de traitement, cela tout en mettant en évidence les contraintes mises en jeu.

Pour ce qui tient à la pragmatique, celle-ci peut être vue comme l'étude de la relation entre signes linguistiques et leur utilisation en contexte, et cela de manière complémentaire à la sémantique, qui concerne l'étude de la relation entre signes linguistiques et entités correspondantes, *référées* par les signes linguistiques, et à la syntaxe, qui étudie la relation entre les signes linguistiques en tant que tels [104]. Autrement dit, la pragmatique étudie la manière dont les signes linguistiques sont influencés par et influencent le contexte de leur *réalisation*, par des sujets humains. Ainsi, les aspects pragmatiques ont un rôle important lorsqu'il s'agit de déterminer le choix des énoncés qui expriment un certain sens *donné*. Ainsi, lorsqu'on construit des machines capables d'interagir de manière linguistique avec l'utilisateur humain, les tentatives d'embarquer des connaissances pragmatiques ont produit plusieurs systèmes où le contexte constitue un aspect essentiel [150]. Cependant, dans ces systèmes déjà munis d'« intelligence artificielle », le contexte est souvent réduit à des bases de connaissances qui spécifient des faits connus par le système ; ces bases de connaissances sont parfois renforcées avec un ensemble limité de capacités inférentielles [128].

Une piste de recherches fructueuse en dialogue homme-machine part souvent des idées de la Théorie des actes de langage (« Speech Act Theory ») de Austin et Searle [156], formalisée de diverses manières [185], [27]. Ainsi, des éléments de pragmatique ont commencé à être pris en compte, surtout au niveau de l'énoncé [172], [19], [164]. Cependant, le contexte ne se réduit pas à la prise en compte des relations sociales entre les partenaires en dialogue, ou des relations entre ces locuteurs et l'environnement ; le contexte comprend aussi la situation de la tâche ou, d'un point de vue linguistique, la situation de dialogue [35]. Les formalisations des structurations discursives en dialogue ont commencé à être utilisées dans la communauté de Génération du langage naturel (« Natural Language Generation ») depuis le début des recherches en ce domaine [147], mais surtout pour des buts monologiques, où on a vite observé que pour exprimer de manière naturelle, *via* des textes, des contenus informationnels (par exemple, extraits des bases de données), on a besoin d'une perspective sur la *cohérence rhétorique* de ces textes.

Le rôle de la pragmatique a été souligné dans plusieurs systèmes pionniers de dialogue homme-machine, comme celui de Jokinen [86], où une version du cadre BDI – « Belief, Desire, Intention » – de Bratman est adaptée pour la planification des énoncés dialogiques. Ces approches ont cependant plusieurs limites : par exemple, du fait que ces formalisations manquent d’une perspective rhétorique, le contenu *linguistique* du dialogue n’est pas formalisé de manière robuste et rigoureuse ; cela revient, entre autres, à ce qu’elles ne sont pas facilement extensibles au dialogue à plusieurs locuteurs, parce qu’elles ne modélisent pas de manière explicite les situations de *négociation* ou de *coopération* entre plusieurs locuteurs, face à une tâche *commune*.

Lorsqu’il s’agit de construire des systèmes de dialogue homme-machine à plusieurs locuteurs, le contexte peut être considéré à plusieurs niveaux [33] :

- étendue du domaine / de la tâche – elle peut être exprimée dans une *ontologie de tâches* où on spécifie à la fois des relations hiérarchiques entre objets et / ou actions dans l’univers de la tâche, et des relations non-hiérarchiques entre ces entités [35] ;
- buts dialogiques des locuteurs – ils peuvent être pris en compte au niveau d’une composante de *contrôle de dialogue*, où un ensemble de stratégies inspirées par la théorie des jeux peut piloter le calcul des intentions communicationnelles ayant pour but la satisfaction des buts des locuteurs [35] ;
- contexte linguistique du dialogue – cela peut s’exprimer *via* une *structure rhétorique*, formalisée et calculée au fur et à mesure que le dialogue progresse ; cette structure rhétorique peut piloter à la fois l’interprétation des énoncés provenus des interlocuteurs humains (insertion dans la structure rhétorique et attribution d’un type d’acte de langage [185], [35]) et leur génération (choix des caractéristiques pragmatiques appropriées pour les formes linguistiques qui expriment les intentions communicationnelles de la machine [136]).

Le contexte est souvent formalisé dans une perspective d’intelligence artificielle, en construisant des ontologies qui recouvrent certains domaines d’application [150], [35]. Ensuite, ces ontologies sont utilisées par un composant spécifique des systèmes de dialogue, le *contrôleur de tâche*, qui construit et manipule des plans pour accomplir les tâches [35], [117].

Le contexte est également pris en compte de plusieurs manières au niveau du contrôle du dialogue : soit dans un cadre inférentiel pur, comme dans les travaux de Sadek [117], soit dans une approche stochastique, comme dans le système de dialogue Communicator de l’Université Carnegie Mellon [117], soit en utilisant des idées empruntées à la théorie de jeux [35]. Quelle que soit l’approche adoptée, elle est en général implantée dans le contrôleur de dialogue [35], [117], [86], [172].

Le contexte linguistique en dialogue peut être abordé à au moins deux niveaux : (i) un niveau **local**, où à chaque énoncé on associe un *type d’acte de langage* et une *forme sémantique* [19], [185], et (ii) un niveau **global**, où on construit et manipule une représentation structurelle de l’ensemble des énoncés qui composent un dialogue ; cette *structure de discours* est en général calculée de manière incrémentale, en adjoignant chaque énoncé récemment produit, à la structure [11]. Chaque nouvel énoncé peut être produit soit par les locuteurs humains – dans ce cas la machine doit les *interpréter* (c’est-à-dire, leur associer un type d’acte de langage et une forme logique, *et* les adjoindre à la structure de discours), soit par la machine, qui doit dans ce cas *générer* ces énoncés (c’est-à-dire, ajouter les formes logiques de ces énoncés à la structure rhétorique *et* construire les formes linguistiques correspondantes). En général, les chercheurs

se sont concentrés soit sur la représentation locale du contexte linguistique en dialogue [172], [164], [19], dans des cadres dérivés de la Théorie des actes de langage [156], [185], soit sur la représentation globale [117], [130], dans des cadres issus de la théorie RST (« Rhetorical Structure Theory ») ou de la SDRT (« Segmented Discourse Representation Theory ») [11], ou selon des approches dérivées de la théorie des jeux [130].

Actuellement, peu d'essais existent de combiner les deux perspectives sur le contexte linguistique ; cela parce que, d'habitude, la perspective locale est adoptée pour l'interprétation des énoncés, en essayant d'associer des types d'actes de langage aux énoncés de l'utilisateur [89], tandis que la perspective globale est adoptée pour la génération des répliques en dialogue ; notre recherche tente de combler ce vide, en proposant un cadre pour la conception des applications de dialogue homme-machine à plusieurs locuteurs [35].

Dans ce contexte, nous avons conçu une approche basée sur la pragmatique à la modélisation du dialogue ; cette démarche tente de fournir des mécanismes rigoureux pour construire la structure rhétorique du dialogue, et pour utiliser celle-ci en génération, pour contrôler certaines caractéristiques des énoncés. L'approche, d'abord conçue et implémentée dans plusieurs applications de dialogue entre une machine et un locuteur humain, a été étendue à certains types de dialogue à *plusieurs* locuteurs, où la machine dialogue simultanément avec un nombre arbitraire d'usagers humains. De plus, nous considérons ce cadre comme un enchaînement d'instances de problèmes de satisfaction de contraintes ; ainsi, on peut préciser ses limites et prédire des améliorations visant à les contourner.

D'un point de vue formel, un problème de satisfaction de contraintes peut être défini en termes d'un ensemble O d'objets : $O = \{o : o = \text{objet}\}$, et d'un ensemble de *contraintes* : $C = \{c : c : O \mapsto S\}$. Chaque objet est défini en termes d'un nombre de variables : $o = \langle X_1, \dots, X_n \rangle$, où chaque variable X_i peut prendre une valeur x_i dans un *domaine* $D_i : o \in D_1 \times \dots \times D_n$. Chaque contrainte est alors une fonction de l'ensemble O d'objets à un ensemble S de *niveaux de satisfaction* ; d'ordinaire, pour des raisons pratiques, $\{0, 1\} \subseteq S \subseteq [0; 1]$, où 0 désigne l'insatisfaction totale, et 1, la satisfaction totale.

Les contraintes peuvent être *dures* (« hard » en anglais) si $S = \{0, 1\}$, et *douces / floues* (« soft » en anglais), si $S = [0; 1]$. Ces contraintes, vues comme fonctions, peuvent être *composées* : pour deux contraintes c_1 et c_2 en C , on dit que $c_1 \circ c_2$ est une contrainte composée¹. Cela est utile pour rendre compte des *solutions* qui satisfont plusieurs contraintes. Une solution à un problème de satisfaction de contraintes, définie en termes des ensembles O , C et S ci-dessus, se représente par un ensemble $O' \subseteq O$ d'objets de O , ainsi que $\forall o' \in O' : \circ_{c \in C} c(o') = \gamma$. Si toutes les contraintes en C sont dures, alors $\gamma = 1$; sinon, $\gamma = \max_{o \in O} (\circ_{c \in C} c(o))$.

Pour des raisons pratiques, on a besoin de donner une signification quantitative à l'opérateur de composition des contraintes ; plusieurs alternatives existent : (i) $\circ \equiv +$, donc les niveaux de satisfaction sont additionnés (ce qui revient, essentiellement, au fait qu'il est suffisant qu'une solution satisfasse *certaines* contraintes (« hard »), et les niveaux de satisfaction sont cumulés)², (ii) $\circ \equiv \min$ (ce qui revient au fait que toutes les contraintes doivent être satisfaites simultanément

¹Il faut cependant remarquer le fait que cette définition de la composition n'est pas tributaire de la définition classique de la composition des fonctions.

²Il faut noter qu'on préfère parfois inverser la correspondance entre les niveaux de satisfaction des contraintes et leur valeur numérique : 0 signifie satisfaction totale et 1, non-satisfaction totale ; ceci étant, l'opérateur d'addition produit des valeurs qui indiquent des *coûts* induits par la non-satisfaction de certaines contraintes [182].

par un objet, afin que ce dernier soit une solution), ou (iii) $\circ \equiv \uparrow$, où par \uparrow nous notons l'opérateur de concaténation ; cette contrainte revient à une application séquentielle des contraintes, comme une série de filtres : les objets qui satisfont une contrainte sont ensuite confrontés à d'autres contraintes, et l'ensemble des solutions est raffiné graduellement. Il est intéressant de constater que les opérateurs de composition (ii) et (iii) ont le même effet sur l'ensemble des solutions retenues ; pourtant, du point de vue algorithmique, les différences concernant le temps de calcul et la mémoire (par exemple, la taille de l'espace de recherche) peuvent être dramatiques.

L'ensemble des solutions à un problème de satisfaction de contraintes peut être raffiné davantage en attribuant des *priorités* aux contraintes, c'est-à-dire, une correspondance $\pi : C \mapsto P$ des contraintes à un ensemble de *niveaux de priorité* $P \subset \mathbb{R}$. π induit ainsi une relation d'ordre sur C , ce qui nous permet de définir clairement l'ensemble des solutions à un problème de satisfaction de contraintes : un objet $o \in O$ est une solution au problème de satisfaction de contraintes, caractérisé par les ensembles O , C , S et P , et par la fonction π , si et seulement si :

$$\exists c \in C : \forall c' \in C : (\pi(c') \leq \pi(c) \Rightarrow c'(o) = 1 \wedge \forall o' \in O \setminus \{o\} : \exists c'' : \pi(c'') \leq \pi(c) \wedge c''(o') = 0).$$

Dans ce cadre, nous pouvons voir comment l'opérateur de concaténation \uparrow (en tant que particularisation de la composition des contraintes) rend bien compte à la fois de la sémantique de l'opérateur min et de l'attribution implicite de niveaux de priorité aux contraintes :

$$\uparrow_{i=1}^N (c_i(o)) \equiv \min_{i=1, \dots, N} (c_i(o)) \wedge (\pi(c_1) \geq \pi(c_2) \geq \dots \geq \pi(c_n)).$$

Une autre classification utile des contraintes regroupe celles-ci en *passives* et *actives*. Les contraintes passives sont vérifiées *a posteriori* afin de choisir l'ensemble des solutions. Ainsi, dans les problèmes de résolution des contraintes passives, on génère d'abord un ensemble de solutions candidats, qu'on teste ensuite face aux contraintes (via un processus de recherche dans l'ensemble des solutions candidats) afin de choisir un sous-ensemble de solutions au problème. Les contraintes actives sont vérifiées *a priori*, c'est-à-dire, avant qu'on calcule les solutions au problème. Ainsi, les solutions sont générées de manière incrémentale par l'application des contraintes ; nous appelons les solutions ainsi obtenues *synthétisées* [182].

Avant d'entrer dans les détails de chaque aspect du contrôle pragmatique de la génération des énoncés en dialogue à plusieurs locuteurs, nous précisons ce que nous entendons par dialogue « à plusieurs locuteurs » ou « multi-locuteurs », en opposition, d'une part, avec le dialogue classique, à deux locuteurs et, d'autre part, avec le dialogue multi-session [35]. Par dialogue à *plusieurs* locuteurs nous entendons un ensemble d'interactions simultanées entre plus que deux locuteurs, où les connaissances ou informations déterminées lors d'une session de dialogue ne sont pas transmises à d'autres sessions ultérieures (dialogue sans mémoire). En revanche, les dialogues multi-session sont, selon Caelen [35], des chaînes alternées de dialogues à deux locuteurs différents, où les informations déterminées dans une session sont transférées aux dialogues ultérieurs, afin qu'il soit possible de résoudre les conflits entre plusieurs utilisateurs qui demandent l'accès à une même ressource.

Ce chapitre, qui résume nos contributions personnelles dans un condensé des chapitres qui suivent, est structuré ainsi : dans la section suivante nous présentons l'architecture de dialogue de Caelen, où nos recherches s'inscrivent ; ensuite, nous mettons en évidence les aspects liés à la génération des énoncés, aussi que les problèmes théoriques et computationnels correspondants ; nous regardons ces derniers en tant qu'instances de problèmes de satisfaction de contraintes.

Finalement, un ensemble de conclusions clôt le chapitre.

1.2 Contexte des recherches

Le contexte de la thèse pourrait être illustré par la figure 1.1, où les modules (encadrés en rectangles) et les structures de données (encadrées en ellipses) représentent les éléments du système à développer.

Sur cette figure 1.1, les significations des notations et des termes utilisés sont les suivantes :

- flèches continues – échanges informationnels entre les modules dans le système de dialogue ;
- flèches en pointillé – échanges de contrôle entre les modules ;
- rectangles – modules (logiciels) de traitement, dénommés selon le texte à l'intérieur ;
- « {...} » – messages échangés entre les modules ;
- « CTRL » – message de contrôle, en cas d'erreur ;
- « RQ » – message de requête ;
- « Parole continue » – signal acoustique « portant » un énoncé véhiculé dans le système ;
- « Schéma sémantique » – formule logique spécifiant (dans la logique des prédicats du premier ordre) le contenu sémantique d'un énoncé ;
- « Chaîne orthographique » – texte d'un énoncé ;
- « Modèle pragmatique » :
 - pour un seul énoncé – spécification logique portée par l'acte de langage (dans le cadre des théories de Austin et Searle) associé ;
 - pour un ensemble d'énoncés – spécification de la structuration discursive / rhétorique (dans le cadre de la théorie SDRT de Asher et Lascarides) de ces énoncés ;
- « Plans / modèles tâche » – ensemble de scripts spécifiant les suites d'actions associées à un ensemble de tâches que la machine est censée exécuter ;
- « Référents tâche » – dictionnaire contenant les mots correspondant aux objets et prédicats dans l'ontologie de tâches ;
- « Intention communicationnelle » – spécification logique (dans la logique des prédicats du premier ordre) d'un énoncé, accompagné par l'acte de langage associé (dans le cadre des théories de Austin et Searle) ;
- « Contour F0 » – représentation de l'ensemble des marques prosodiques (dont le contour de la fréquence fondamentale) pour la réalisation acoustique (prononciation) d'un énoncé ;
- « RECONNAISSANCE » – module (logiciel) réalisant la conversion d'un signal acoustique de parole, vers le texte correspondant, pour un énoncé ;
- « ANALYSE SEMANTIQUE » – module (logiciel) réalisant la conversion d'un texte vers la représentation logique (dans le cadre de la logique des prédicats du premier ordre) ; il utilise actuellement (cf. [122]) le formalisme des graphes conceptuels, conçu par Sowa, et une grammaire sémantique définie pour un domaine donné ;
- « INTERPRETATION PRAGMATIQUE » – module (logiciel) ajoutant à la représentation sémantique d'un énoncé, l'acte de langage (dans le cadre des théories de Austin et Searle) associé et, pour plusieurs énoncés, construisant la structure discursive (dans le cadre de la théorie SDRT) qui relie ces derniers ; il utilise, au niveau d'un énoncé, la taxinomie

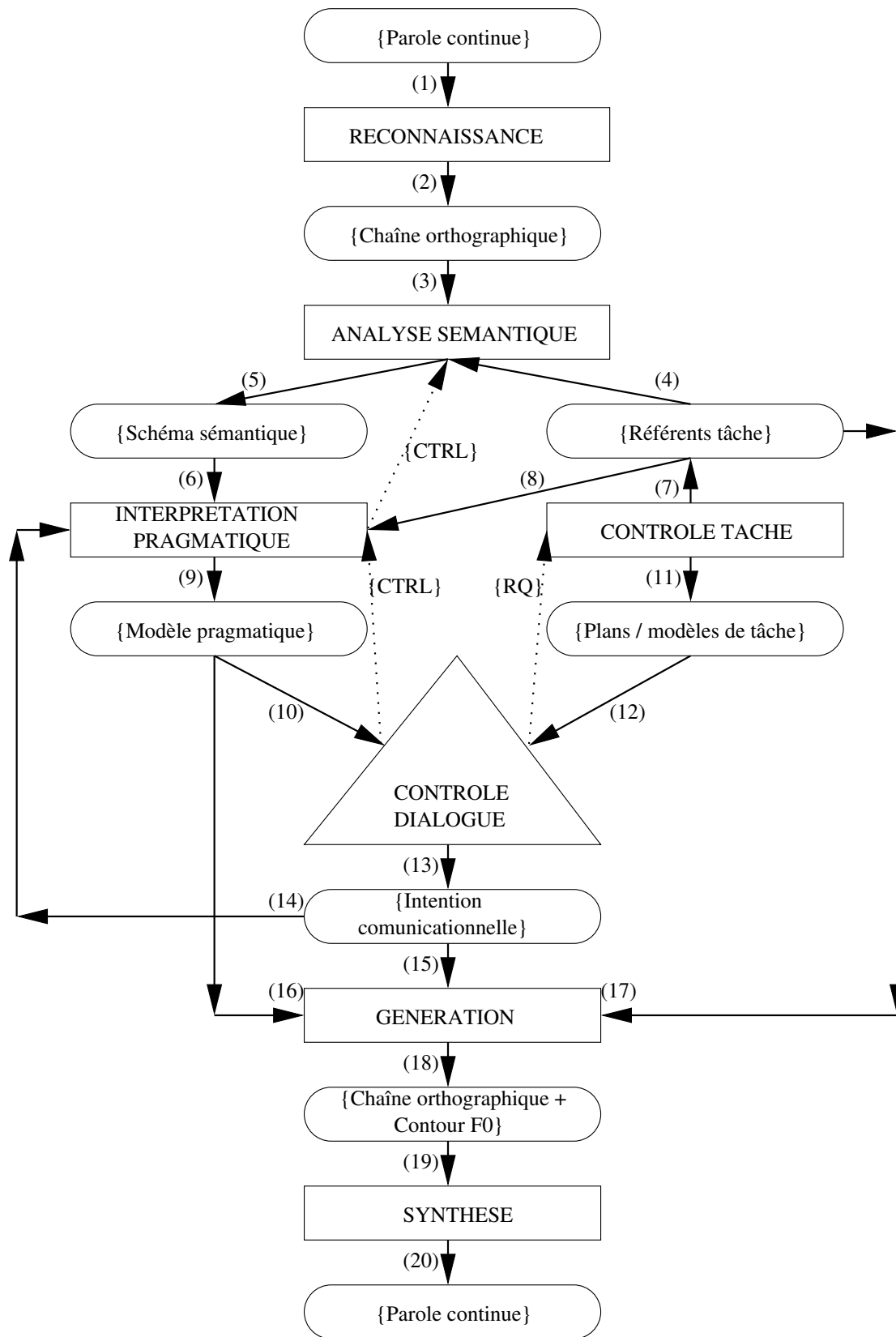


FIG. 1.1 – Le contexte des recherches

des actes de langage proposée par Caelen et, au niveau de la structuration rhétorique, l'ensemble des relations rhétoriques choisi par Xuereb et Caelen, dans le cadre de la SDRT [35] ;

- « CONTROLE TACHE » – module (logiciel) gérant un ensemble de plans / modèles de tâche et l'ensemble des référents de tâche constituant l'ontologie du domaine d'utilisation ; il utilise le cadre défini par Caelen et Villaseñor pour l'apprentissage et la gestion des plans / modèles de tâche, et sur les grammaires sémantiques de Sowa pour représenter l'ontologie de tâches, de manière conjointe avec le module d'analyse sémantique ;
- « CONTROLE DIALOGUE » – module (logiciel) gérant le déroulement du dialogue entre l'utilisateur humain et la machine ; il utilise le cadre formel conçu par Caelen, s'appuyant sur une logique dialogique dérivée de la logique des prédicats du premier ordre et augmentée avec des opérateurs modaux, fournissant en sortie l'intention communicationnelle pour une réplique en dialogue (une réplique représente un seul énoncé ou un ensemble d'énoncés produits par le même locuteur dans le même tour de parole) ;
- « GENERATION » – module (logiciel) réalisant la conversion d'une intention communicationnelle, vers la chaîne orthographique et les contours prosodiques correspondants ;
- « SYNTHESE » – module (logiciel) réalisant la conversion d'une chaîne orthographique accompagnée par un contour prosodique, vers un signal acoustique de parole continue ;
- « (1) », ..., « (20) » – nombres attachés aux flèches continues, indiquant le flux logique de l'information dans le système de dialogue.

Toujours s'appuyant sur la figure 1.1, nous détaillons maintenant l'algorithme général de fonctionnement de l'architecture de dialogue présentée ; donc, le *modus operandi*, pour un couple de deux tours de parole provenant de l'utilisateur et de la machine respectivement, est :

1. Le module de RECONNAISSANCE **reçoit** le signal de parole continue ;
2. Le module de RECONNAISSANCE **fournit** la chaîne orthographique ;
3. Le module d'ANALYSE SEMANTIQUE **reçoit** la chaîne orthographique ;
4. Le module d'ANALYSE SEMANTIQUE **lit** l'ensemble des référents de tâche ;
5. Le module d'ANALYSE SEMANTIQUE **fournit** le schéma sémantique ;
6. Le module d'INTERPRETATION PRAGMATIQUE **reçoit** le schéma sémantique ;
7. Le module de CONTROLE TACHE **produit** l'ensemble des référents de tâche ;
8. Le module d'INTERPRETATION PRAGMATIQUE **lit** l'ensemble des référents de tâche ;
9. Le module d'INTERPRETATION PRAGMATIQUE **fournit** le modèle pragmatique ;
10. Le module de CONTROLE DIALOGUE **reçoit** le modèle pragmatique ;
11. Le module de CONTROLE TACHE **produit** les plans / modèles de tâche ;
12. Le module de CONTROLE DIALOGUE **lit** les plans / modèles de tâche ;
13. Le module de CONTROLE DIALOGUE **fournit** l'intention communicationnelle ;
14. Le module d'INTERPRETATION PRAGMATIQUE **reçoit** l'intention communicationnelle ;
15. Le module de GENERATION **reçoit** l'intention communicationnelle ;

16. Le module de GENERATION **reçoit** le modèle pragmatique ;
17. Le module de GENERATION **lit** l'ensemble des référents de tâche ;
18. Le module de GENERATION **fournit** la chaîne orthographique et le contour de la fréquence fondamentale ;
19. Le module de SYNTHÈSE **reçoit** la chaîne orthographique et le contour de la fréquence fondamentale ;
20. Le module de SYNTHÈSE **fournit** le signal acoustique de parole continue.

Dans la description de l'algorithme général d'opération nous avons employé plusieurs verbes (marqués en gras) :

- Pour les opérations associées aux flèches continues *incidentes* aux modules :
 - « **recevoir** » : pour l'information incidente associée à un échange dialogique momentané, dynamique ;
 - « **lire** » : pour l'information incidente associée à une interrogation d'une base de données statique ;
- Pour les opérations associées aux flèches continues *émergentes* aux modules :
 - « **fournir** » : pour l'information émergente associée à un échange dialogique momentané, dynamique ;
 - « **produire** » : pour l'information émergente associée à une mise à jour d'une base de données statique.

Ainsi, les étapes de traitement pour deux tours de parole, de l'utilisateur et respectivement de la machine en dialogue sont les suivantes³ :

1. L'utilisateur *U* produit l'énoncé oral 'Bonjour, je voudrais un livre sur Jules Verne !' ;
2. Le module de RECONNAISSANCE (qui réalise un traitement d'*entrée* dans le système de dialogue) convertit la réalisation acoustique de l'énoncé vers la chaîne orthographique 'bonjour je voudrais un livre sur jules verne' ;
3. Le module de COMPREHENSION (en fait, d'ANALYSE SEMANTIQUE) (qui réalise un traitement d'*entrée* dans le système de dialogue) convertit la chaîne de caractères vers une formule logique constituant le but communicationnel de l'énoncé ; une telle formule pourrait se représenter de manière plutôt informelle ainsi :
 $\exists ? \text{livre} \wedge \text{livre}(\text{ sujet } = ' \text{Jules Verne}') ;$
4. Le module d'INTERPRETATION (PRAGMATIQUE) (qui réalise un traitement d'*entrée* dans le système de dialogue) détermine que le type d'acte de langage correspondant à l'énoncé est *Faire Savoir* et que le degré de force illocutoire est *moyen* (plus précisément, d'une valeur de -1 sur une échelle de -2 à $+2$ – cf. le chapitre 7) ;
5. Le module de CONTROLE DIALOGUE (qui réalise un traitement de *gestion* dans le système de dialogue) détermine que l'utilisateur a employé une stratégie *directive* pour produire l'énoncé en question ;

³Pour les deux tours de parole qui restent, les traitements sont semblables à ceux juste décrits.

6. Le module de CONTROLE DIALOGUE interroge le module de CONTROLE TACHE (qui réalise aussi un traitement de *gestion* dans le système de dialogue) et produit, par conséquent, l'intention communicationnelle du type
 $\exists \text{ livre} \wedge \text{livre}(\text{ sujet} = \text{'Jules Verne'}) \wedge \text{ sujet} = \text{'vie'} \vee \text{ sujet} = \text{'oeuvre'}$;
7. Le module de CONTROLE DIALOGUE choisit une stratégie *coopérative* et un acte de langage *Faire Pouvoir* pour communiquer l'intention produite à l'étape antérieure ;
8. Le module de GENERATION (qui réalise un traitement de *sortie* dans le système de dialogue) calcule le degré de force illocutoire (*faible*) pour l'intention communicationnelle à produire sous forme linguistique par la machine ;
9. Le module de GENERATION produit la chaîne orthographique 'Bonjour, voudriez-vous en un sur la vie ou sur l'oeuvre de Jules Verne ?'

On observe ainsi que les traitements réalisés ne sont pas symétriques (entre l'entrée et la sortie) et correspondent à la description du cadre choisi. Il faut cependant noter que ce cadre a été initialement conçu pour le dialogue classique, à deux locuteurs, pour être ensuite étendu au dialogue multi-session. Néanmoins, le cadre est encore à étendre (surtout au niveau du contrôle du dialogue) au dialogue multi-locuteurs.

1.3 Contrôle pragmatique de la génération des énoncés en dialogue à plusieurs locuteurs

1.3.1 Perspective rhétorique sur le dialogue à plusieurs locuteurs

Nous considérons que le dialogue homme-machine est une activité conjointe, où les locuteurs sont motivés au moins pour maintenir une interaction cohérente, afin de réaliser certains buts (cela n'exclut pas les conflits). Une motivation plutôt « fondamentale » pour cette perspective est présentée en [35] (voir notamment les chapitres 1, 3, 5 et 7), mais dans cet ouvrage le dialogue à plusieurs locuteurs n'est pas envisagé⁴. Cependant, notre traitement du dialogue à plusieurs locuteurs ne concerne que les aspects discursifs, notamment le calcul d'une représentation rhétorique du dialogue et la manière dont celle-ci peut piloter le choix des énoncés pertinents par rapport au contexte de l'interaction.

Tout d'abord, nous définissons le dialogue homme-machine à plusieurs locuteurs comme une interaction où la machine est impliquée dans des dialogues *simultanés* avec plusieurs locuteurs ; la machine doit garder une vue cohérente sur ces dialogues. Dans notre perspective (présentée tout au long de cette thèse), la représentation que la machine (se) construit sur les dialogues est exprimée comme un ensemble de structures segmentées de discours (« Segmented Discourse Representation Structures » – SDRS), au sens de la théorie SDRT [11].

Nous posons que le dialogue à plusieurs locuteurs peut être regardé comme une superposition de dialogues non-disjoints à deux locuteurs, où un locuteur (humain ou artificiel) peut parler simultanément à plusieurs locuteurs qui, à leur tour, peuvent faire de même. Donc, une situation de dialogue à plusieurs locuteurs peut être vue comme un ensemble de dialogues (éventuellement, partiellement chevauchants), entre des paires de locuteurs. A un niveau discursif, cela

⁴Dans *op. cit.*, seuls les dialogues multi-session sont considérés.

est formalisé comme un ensemble de structures SDRS qui reflètent, chacune d'entre elles, un dialogue à deux locuteurs (voir chapitre 5). Le chevauchement éventuel entre les dialogues à deux locuteurs revient au fait qu'il existe des énoncés qu'un locuteur adresse simultanément à plusieurs partenaires de dialogue.

De son côté, la machine peut soit participer directement au dialogue (dans ce cas, nous disons que la machine est un « locuteur impliqué »), soit seulement observer d'autres (paires de) locuteurs qui parlent (dans ce cas, nous disons que la machine est un « locuteur extérieur »). Cette distinction, importante lorsque nous nous intéressons à la mise à jour des structures discursives quand un nouvel énoncé doit être produit par la machine, est discutée de manière formelle dans le chapitre 5.

De plus, pour chacune de ces situations, la formalisation est régie par deux postulats qui résultent du fait que les dialogues sont coopératifs (dans le sens des maximes de Grice [104]) :

- *Vue commune* : Deux locuteurs quelconques, impliqués dans un dialogue, gardent la même vue (perspective) sur celui-ci ; cela paraît de bon sens si on suppose que les dialogues sont coopératifs et que les locuteurs se comportent selon leur statut – par exemple, la machine comme un employé d'une institution et les locuteurs humains comme des clients de cette institution ;
- *Réflexivité* : Pour trois locuteurs quelconques, dont deux impliqués dans un dialogue et le troisième les observant, la vue de ce dernier locuteur sur le dialogue est invariante par rapport à l'ordre dans lequel les deux locuteurs sont considérés, c'est-à-dire, un dialogue entre un locuteur *A* et un locuteur *B* est vu de la même manière que le dialogue entre le locuteur *B* et le locuteur *A*, par un locuteur *C*.

De plus, chacune de ces deux « vues » précisées ci-dessus revient à deux situations : (i) plusieurs dialogues, où la machine est soit impliquée, soit observateur, ont lieu en parallèle et il n'y a pas de véritable interaction à plusieurs locuteurs (les dialogues sont disjoints), et (ii) les dialogues ayant lieu simultanément ne sont pas disjoints, donc une véritable interaction à plusieurs locuteurs existe. Ces différences sont aussi formalisées, à un niveau rhétorique, dans le chapitre 5.

1.3.2 Structuration rhétorique en génération des énoncés

Dans la conception d'une composante de structuration rhétorique pour la génération des énoncés en dialogue à plusieurs locuteurs, nous avons suivi une approche incrémentale, c'est-à-dire, en précisant dans un premier temps une telle composante pour le dialogue à deux locuteurs, pour ensuite étendre cette dernière au dialogue à plusieurs locuteurs. Par conséquent, nous allons d'abord décrire la composante de structuration rhétorique pour le cas du dialogue à deux locuteurs.

Cas du dialogue à deux locuteurs

Afin de pouvoir spécifier un mécanisme computationnel pour mettre à jour la structure de discours d'un dialogue « classique » (à deux locuteurs), lorsque la machine est en train de générer une réplique, nous avons choisi la SDRT comme un point de départ. Ce choix est dû au fait que la SDRT propose un ensemble de relations rhétoriques d'une granularité assez fine, quoique non

restrictive ; ces relations concernent, de manière explicite, le dialogue, ainsi que le monologue [11].

Ainsi, des 35 relations rhétoriques proposées par la SDRT, nous en avons choisi 17 (considérées comme pertinentes dans un contexte de dialogue, à partir d'une étude empirique sur un corpus de dialogues [35]), que nous avons d'abord regroupées en deux catégories :

1. relations rhétoriques **monologiques** – *Alternation, Background, Consequence, Elaboration, Narration, Contrast et Parallel* ;
2. relations rhétoriques **dialogiques** – *Q-Elab, IQAP, P-Corr, P-Elab, Background_q, Elab_q, Narration_q, QAP, ACK et NEI*.

Afin d'établir des moyens computationnels pour mettre à jour ces structures de discours (construites de manière récursive comme des graphes, où les types des relations rhétoriques sont les arêtes et les étiquettes des énoncés sont des nœuds), les sémantiques des relations rhétoriques ont été formellement spécifiées en utilisant un formalisme logique du premier ordre. Ce formalisme est *le même* pour les relations rhétoriques *et* pour les sémantiques des énoncés. Néanmoins, il existe, comme dans la SDRT d'origine, une séparation entre les sémantiques des énoncés et celles des relations rhétoriques (en ce qui concerne l'accessibilité des informations entre ces deux). Pourtant, à la différence de la SDRT, où cette séparation est réalisée en utilisant des *formalismes logiques* différents pour les deux types d'objets (relations rhétoriques et énoncés), dans notre approche le formalisme en tant que tel est le même, mais les *bases de connaissances* sont différentes pour les deux types d'objets. Ainsi, tandis que les sémantiques des énoncés sont essentiellement exprimées *via* des prédicats et des objets dans une *ontologie de tâches* qui est spécifique au domaine des dialogues traités par le système, les sémantiques des relations rhétoriques sont exprimées *via* des prédicats que nous appelons *discursifs*. Les objets des formes logiques exprimant les sémantiques des relations rhétoriques sont des énoncés.

Les sémantiques des relations rhétoriques sont donc émulées dans une logique du premier ordre qui s'exprime à travers un ensemble de prédicats discursifs, tandis que le processus de mise à jour des structures rhétoriques est spécifié *via* un algorithme (décrit de manière étendue dans les chapitres 3 et 4) qui, essentiellement, pour chaque type de relation rhétorique (des 17 considérées), prend la forme logique qui exprime l'intention communicationnelle d'un énoncé que la machine est censée produire sous forme linguistique, aussi que les formes logiques des énoncés (ou constituants discursifs – c'est-à-dire, des sous-structures SDRS) antérieurs dans le dialogue, afin de dériver, de manière compositionnelle, une forme logique (construite à partir des sémantiques de la relation rhétorique, dont les arguments – c'est-à-dire, les énoncés – sont éclatés à leur tour dans leurs formes logiques), dont la valeur de vérité est évaluée (selon les connaissances spécifiées dans l'ontologie de tâches et dans l'ensemble des prédicats discursifs).

Il n'en est pas moins vrai que cette procédure plutôt simpliste de mise à jour des structures de discours est coûteuse du point de vue computationnel (puisque chaque type de relation rhétorique possible doit être vérifié, pour chaque énoncé antérieur dans le dialogue). Par conséquent, nous avons proposé un mécanisme pour réduire, d'une part, ce temps de calcul et obtenir, d'une autre part, des structures rhétoriques plus fidèles aux prédictions linguistiques. Le problème ici est que parfois les sémantiques des relations rhétoriques sont trop permissives, au sens où même des relations rhétoriques qui ne peuvent pas exister (selon les prédictions linguistiques) sont

prouvées vraies par l'algorithme. Ainsi, nous avons ajouté des contraintes sous forme de types d'actes de langage associés aux énoncés. Plus précisément, à partir des constats empiriques qui ont montré que les paires d'actes de langage limitent l'ensemble des relations rhétoriques qui peuvent relier les énoncés « porteurs » de ces actes [35], nous avons amélioré l'algorithme de structuration rhétorique avec des règles qui limitent, pour une paire d'actes de langage donnée, l'ensemble des relations rhétoriques qui peuvent relier les énoncés qui réalisent ces actes. Ces règles, présentées en détail dans le chapitre 3, sont basées sur des prescriptions du type :

Un énoncé qui réalise un acte « faire-faire-savoir » (F^{FS} , requête d'informations), et un énoncé ultérieur qui réalise un acte « faire-savoir » (F^S , offre d'informations), peuvent être reliés seulement par des relations rhétoriques du type *QAP*, *IQAP* et *P-Elab*.

Une liste complète des paires d'actes de langage, avec les relations rhétoriques qui peuvent relier les énoncés réalisant ces actes, est donnée, accompagnée par des exemples de dialogues, en [35].

Le processus de structuration rhétorique, en dialogue à deux locuteurs, consiste essentiellement à parcourir trois étapes : (i) un filtrage local « pragmatique » des relations rhétoriques *potentiellement* valides, *via* les actes de langage (des contraintes induites par les paires d'actes de langage sur les relations rhétoriques qui peuvent relier ces actes), (ii) un filtrage local « sémantique » (via des techniques de démonstration de théorèmes [63]) d'un sous-ensemble de relations rhétoriques valides, (iii) un filtrage contextuel « syntaxique » sur la sortie de l'étape antérieure. Ce dernier filtrage suppose, à son tour, trois étapes : (a) chaque relation rhétorique est exprimée en termes d'une sémantique du premier ordre, définie à partir de prédicats discursifs ; (b) chaque sémantique de ce type est convertie dans un programme logique ; (c) chaque programme logique ainsi obtenu est converti à son tour dans une grammaire abstraite d'attributs (« Abstract Attribute Grammar » – AAG). Ainsi, nous prenons en compte la structure entière du discours, dans une perspective plutôt « syntaxique » ; le critère de filtrage réside dans l'unifiabilité de tous les ensembles de restrictions qui correspondent à une variable globale dans la grammaire AAG de tout le discours.

En conclusion pour la structuration rhétorique en dialogue à deux locuteurs, nous insistons sur le fait qu'il y a plusieurs instances de problèmes de satisfaction de contraintes : tout d'abord, la *réduction des problèmes* est appliquée lorsque le calcul d'une structure rhétorique cohérente est remis à un processus de filtrage en cascade : filtrage pragmatique à base d'actes de langage, filtrage sémantique et, finalement, filtrage contextuel « syntaxique ». Chacune de ces étapes de filtrage est, à son tour, un problème de satisfaction de contraintes : dans le filtrage pragmatique, des contraintes *passives dures* sur l'ensemble des relations rhétoriques autorisées guident le processus de *recherche* dans l'ensemble des types de relations rhétoriques ; dans le filtrage sémantique, des contraintes *passives dures* sur les sémantiques des relations rhétoriques et des énoncés (en rapport avec l'ontologie des tâches et avec les prédicats discursifs) guident le processus de *recherche*, afin de raffiner davantage l'ensemble des relations rhétoriques potentiellement cohérentes ; enfin, le filtrage contextuel s'appuie sur des contraintes *passives dures* sur les éléments des ensembles de restriction des variables globales dans la grammaire AAG du discours, afin de guider le processus de *recherche*, qui retient un ensemble même plus petit de relations rhétoriques cohérentes.

Extensions vers le dialogue à plusieurs locuteurs

Les dialogues orientés service qui font l'objet de nos recherches peuvent être regroupés en deux catégories, selon le rôle joué par la machine :

- A. Dialogue où la machine (par exemple, un bibliothécaire) parle simultanément à plusieurs clients qui ne parlent pas entre eux ; dans ce cas, le contexte rhétorique qui est pris en compte par la machine se réduit à l'ensemble des structures SDRS qui correspondent aux dialogues entre des paires de locuteurs, dont un est toujours la machine ;
- B. Dialogue où la machine est un « client » qui parle à un utilisateur unique (par exemple, un bibliothécaire), que nous appelons « utilisateur central », tout en prenant en compte les conversations entre cet utilisateur et les autres interlocuteurs.

Dans le cas **A.** la machine est toujours un locuteur « impliqué », tandis que dans le cas **B.**, la machine est soit un locuteur impliqué, soit juste en train de faire la transition d'un locuteur « extérieur » vers un locuteur « impliqué » – après que le locuteur « central » a parlé à un autre locuteur.

D'une manière semblable au cas du dialogue à deux locuteurs, en dialogue à plusieurs locuteurs nous avons aussi proposé un algorithme pour mettre à jour la structure de discours (en fait, ici il s'agit de la réunion de plusieurs structures rhétoriques qui rendent compte des dialogues entre des paires de locuteurs). Fidèles au sujet et à la portée de nos recherches, nous considérons seulement le cas où la machine est en train de générer un tour de parole. Cependant, le cadre formel développé pourrait être également utilisé en interprétation des discours, pourvu que nous disposions d'un analyseur sémantique, qui calcule la représentation sémantique des énoncés provenus des utilisateurs.

Dans le cas **A.**, l'algorithme pour mettre à jour la structure de discours est plutôt semblable au cas du dialogue à deux locuteurs, en ce qu'une forme logique censée être générée (sous forme linguistique) est ajoutée aux structures de discours qui impliquent la machine et les destinataires des énoncés dans ce tour de parole-là⁵. Ainsi, il y a essentiellement autant de processus de mise à jour de structure de discours correspondante aux dialogues à deux locuteurs, que des locuteurs qui sont les destinataires d'un tour de parole précisé (voir chapitre 5).

Dans le cas **B.**, l'algorithme pour la mise à jour des structures de discours est plus élaboré, au sens où une structure de discours *artificielle* est d'abord calculée ; ensuite, cette structure est mise à jour comme dans le cas du dialogue à deux locuteurs. Dans ce cas, il ne peut y avoir qu'un destinataire d'un tour de parole. Le calcul de la structure de discours artificielle censée être mise à jour s'appuie sur deux idées de base : (i) la prise en compte des dialogues auxquels les autres locuteurs (différents de la machine) ont participé de manière active (c'est-à-dire, pas seulement en tant qu'observateurs) et où le locuteur « central » était partenaire, et (ii) le regroupement des relations rhétoriques dans deux classes, relations de *confirmation* et relations de *contradiction* :

- relations de **confirmation** – *Q-Elab*, *P-Elab*, *Elab_q*, *Narration_q*, *QAP*, *ACK*, *IQAP*, *Background*, *Consequence*, *Elab*, *Narration* et *Parallel* ; via ces relations rhétoriques, l'énoncé courant (qui apparaît comme le second argument des relations) ne met pas en question les énoncés antérieurs (qui apparaissent comme le premier argument de ces relations) ;

⁵Si, dans un tour de parole, il y a plusieurs énoncés, adressés à des locuteurs différents, alors ce tour est éclaté en autant de tours distincts que des destinataires ; chaque nouveau tour contient seulement des énoncés contigus, destinés à un locuteur.

- relations de **contradiction** – *P-Corr*, *Background_q*, *NEI*, *Alternation* et *Contrast* ; via ces relations, l'énoncé courant met en question (ou contredit) les énoncés antérieurs auxquels l'énoncé courant est relié (via ces relations de contradiction).

Essentiellement, l'algorithme de mise à jour de la structure rhétorique, dans le cas **B.**, commence avec la représentation rhétorique du dialogue entre la machine et l'utilisateur « central » ; cette structure est ensuite augmentée en prenant en compte les autres conversations (à deux locuteurs) où l'utilisateur « central » participe, ainsi que les types des relations rhétoriques déjà calculées (via la procédure de mise à jour de la structure rhétorique en dialogue à deux locuteurs). Ainsi, en fonction du type de ces relations rhétoriques (de confirmation ou de contradiction), l'ensemble des énoncés candidats (c'est-à-dire, énoncés qui pourraient être reliés, via des relations rhétoriques, à l'énoncé censé être produit par la machine) est filtré (voir chapitre 5). Ensuite, l'ensemble des énoncés restants, avec les relations rhétoriques qui les relie (relations calculées, à leur tour, par la composante d'*interprétation pragmatique*) représentent la SDRS artificielle mentionnée ci-dessus. Enfin, cette structure rhétorique est mise à jour avec l'énoncé censé être produit par la machine. L'algorithme est présenté, discuté et illustré de manière détaillée dans le chapitre 5 de ce document.

De plus, dans le chapitre 5 nous montrons comment un dialogue typique entre une machine et plusieurs locuteurs peut être modélisé comme une alternation des cas **A.** et **B.** mentionnés ci-dessus : dans un dialogue typique la machine est, au début, dans la situation **A.**, pour ensuite commuter occasionnellement en situation **B.**. Le processus de commutation automatique entre les situations **A.** et **B.** est géré en employant une « relation de discours » supplémentaire, *MP*, qui signifie « Situation de dialogue à plusieurs » (« Multi-party »), définie comme suit (voir chapitre 5) :

Une relation *MP* relie deux énoncés si et seulement si son argument gauche est le dernier énoncé d'un locuteur, son argument droit est le premier énoncé d'un autre locuteur (qui parle après le premier), et aucun de ces deux énoncés n'a été produit par la machine.

Ainsi, de manière implicite la machine se trouve dans la situation de dialogue **A.** et, si, dans un tour de parole, il y a une relation *MP*, alors elle commute en situation **B.** ; cette décision est prise au niveau de chaque tour de parole.

Nous pouvons donc conclure que la structuration rhétorique en dialogue à plusieurs locuteurs est une instance d'un problème de satisfaction de contraintes, où les algorithmes imposent des contraintes dures sur la structure de discours ; ces contraintes sont *actives* au sens où elles sont appliquées *a priori*, afin de guider la « synthèse » de la structure rhétorique. Ainsi, nous avons affaire à un problème de satisfaction de contraintes où des contraintes *dures* et *actives* sont mises en valeur lors de la *synthèse* des solutions [182].

1.3.3 Contrôle rhétorique des phénomènes linguistiques en génération des énoncés

La structuration rhétorique pour la production des énoncés en dialogue à plusieurs locuteurs peut être utilisée pour contrôler certains phénomènes concernant le passage de l'intention communicationnelle (exprimée sous forme logique) – qui représente le *sens* d'un énoncé –, à la *forme*

linguistique proprement dite de cet énoncé. Ainsi, la structure du discours peut être utilisée pour piloter certaines décisions concernant l'ellipse sémantique (voir chapitre 6), le degré de puissance de la force illocutoire (c'est-à-dire, du type d'acte de langage) réalisé *via* l'énoncé (voir chapitre 7), ou le choix lexical des connecteurs de discours (voir chapitres 8 et 9). Dans les sections suivantes nous allons présenter de manière concise les mécanismes qui sont en arrière-plan de ces décisions.

Avant cela, nous allons toutefois aussi préciser d'autres éléments sémantiques et pragmatiques qui participent au processus de prise de décision, concernant les phénomènes soulignés ci-dessus.

Questions pragmatiques : engagements publics et structure de discours

L'idée de relier la structure de discours (exprimée, par exemple, dans le cadre de la SDRT, comme c'est le cas dans notre approche) aux *engagements publics* des locuteurs n'est pas nouvelle, car ce lien a été déjà exploité en interprétation des dialogues, afin de remplacer les approches à base d'intentions (par exemple, les logiques BDI – « Belief, Desire, Intention » dans la filiation de Bratman [27]) par une attitude plutôt « agnostique » devant la communication, ayant à la base seulement les choses assumées ouvertement par les locuteurs [112]. Ainsi, le point de départ dans la combinaison des deux vues sur l'interaction (celle « discursive », plutôt sémantique et structurelle, et celle à base d'engagements, plutôt tributaire de la théorie des jeux et prédictive [73]) est l'observation que les relations rhétoriques (par exemple, dans le cadre de la SDRT) déterminent la mise à jour des listes (ensembles) d'engagements des locuteurs. Ce processus, par lequel l'ensemble des listes d'engagements des locuteurs est mis à jour *via* la structure rhétorique représentant le dialogue, a été formalisé en adaptant l'approche de Muller, Maudet et Prévot [112] à notre reformulation (d'un fragment) de la SDRT pour la génération du langage (voir le chapitre 7).

Ainsi, la mise à jour des listes d'engagements, pour un locuteur donné, a lieu différemment, en fonction du contexte de réalisation d'un énoncé (courant) : en dialogue (c'est-à-dire, comme une conséquence d'un énoncé précédent d'un autre locuteur, ou afin de déterminer la production d'un énoncé par un autre locuteur), ou en monologue (c'est-à-dire, comme une suite à un énoncé précédent du même locuteur, dans le même tour de parole, ou comme une prémisse d'un énoncé ultérieur du même locuteur, dans le même tour de parole) :

1. En situation de **monologue**, pour un locuteur donné, ayant produit deux énoncés, reliés par une relation rhétorique monologique (du troisième ordre), sa liste d'engagements est mise à jour en deux étapes, en ajoutant d'abord le contenu sémantique du premier énoncé (exprimé sous forme logique, *via* des prédicats dans l'ontologie de tâches), pour ensuite ajouter le contenu sémantique du deuxième énoncé *et* la sémantique de la relation rhétorique reliant ces deux énoncés (cette sémantique est exprimée en termes de prédicats discursifs) ;
2. En situation de **dialogue**, pour deux locuteurs différents, chacun ayant produit un énoncé, nous avons un des cas suivants :
 - (a) une relation rhétorique (dialogique) de confirmation relie ces deux énoncés ; dans ce cas, la liste d'engagements du locuteur qui a produit le premier énoncé est mise à

jour avec la sémantique de cet énoncé, tandis que l'émetteur du deuxième énoncé met à jour sa propre liste d'engagements avec les sémantiques des deux énoncés et de la relation rhétorique qui les relie ;

- (b) une relation rhétorique (dialogique) de contradiction relie ces deux énoncés ; dans ce cas, la liste d'engagements du locuteur ayant produit le premier énoncé est mise à jour comme dans le cas précédent, tandis que l'émetteur du deuxième énoncé met à jour sa liste d'engagements avec la sémantique du deuxième énoncé, la sémantique *niée* du premier énoncé et la sémantique de la relation rhétorique qui relie ces deux énoncés.

Dans ce même contexte, afin de formaliser la manière dont la liste d'engagements pour l'émetteur du premier énoncé est mise à jour *après* que l'autre locuteur a produit le deuxième énoncé, on doit attendre un troisième énoncé, provenant de l'émetteur du premier énoncé ; ensuite, en fonction du type de relation rhétorique dialogique qui relie le troisième énoncé au deuxième, et du type de relation rhétorique reliant le deuxième énoncé au premier, on a une des situations suivantes :

1. la liste d'engagements du locuteur ayant produit le premier (et ensuite, le troisième) énoncé est mise à jour avec la sémantique du deuxième énoncé, *pourvu que* des relations rhétoriques dialogiques de confirmation relient le troisième énoncé au deuxième, et ce deuxième énoncé au premier ;
2. la liste d'engagements de l'émetteur du premier (et ensuite, du troisième) énoncé est mise à jour en *enlevant* la forme sémantique du premier énoncé et en ajoutant la forme sémantique du deuxième énoncé, *pourvu qu'*une relation rhétorique dialogique de confirmation relie le troisième énoncé au deuxième, et qu'une relation de contradiction relie ce deuxième énoncé au premier ;
3. la liste d'engagements du locuteur ayant produit le premier énoncé (et ensuite, le troisième énoncé) est mise à jour avec la sémantique *niée* du deuxième énoncé, *pourvu qu'*une relation rhétorique dialogique de contradiction relie ce deuxième énoncé au troisième.

Même si à chaque fois qu'un locuteur produit un énoncé qui est relié à des énoncés antérieurs en dialogue, *via* certaines relations rhétoriques, les sémantiques de ces relations rhétoriques sont ajoutées à la liste d'engagements du locuteur [112], dans le mécanisme pour le calcul du degré de force illocutoire des énoncés, nous ignorons parfois (lorsque cela ne nuit pas au traitement) l'ajout de ces sémantiques aux listes d'engagements des locuteurs, en ne retenant que les sémantiques (éventuellement niées) des énoncés (voir chapitre 7).

Modélisation des connaissances des interlocuteurs : référentiel de faits

Les connaissances que les locuteurs acquièrent et développent lorsqu'ils s'engagent en dialogue avec d'autres locuteurs (dont la machine en est un) sont dérivées de leurs listes d'engagements. Ainsi, ces connaissances sont modélisées *via* une **pile** de prédicats (dans l'ontologie de tâches), qui expriment, du point de vue sémantique, les énoncés produits en dialogue. La structure de pile est motivée par le fait que des références *implicites* faites par un locuteur vers

des concepts introduits plus *récemment* dans le discours, sont plus faciles à être compris par les allocutaires.

La pile des prédicats est donc obtenue, pour chaque locuteur, à partir de sa liste d'engagements, *via* deux processus : (i) élimination des sémantiques des relations rhétoriques qui relient les énoncés produits en dialogue, et (ii) la *la réification* des prédicats spécifiques à la tâche, qui « composent » les sémantiques des énoncés (voir le chapitre 6). La réification des prédicats consiste à éliminer les quantificateurs existentiels qui s'attachent aux variables non-typées dont la valeur est attribuée ensuite ; ces variables sont ainsi directement attachées à leurs valeurs. Enfin, ces variables sont éliminées, en unifiant les prédicats qui précisent leurs rôles sémantiques, avec leurs valeurs. Ainsi, la pile de prédicats d'un locuteur (pile qui représente l'ensemble des faits auxquels le locuteur s'est engagé) contient :

- les prédicats affirmés par le locuteur (y compris les prédicats affirmés sous forme de négation) ;
- prédicats affirmés par la machine (ou un autre locuteur, en dialogue à plusieurs locuteurs) et *confirmés*, de manière explicite ou implicite – voir chapitre 6, par le locuteur dont la pile de prédicats est en question.

Un prédicat est confirmé de manière explicite *via* un énoncé relié par une relation rhétorique de confirmation à l'énoncé qui contient ce prédicat, tandis que la confirmation implicite est réalisée *via* l'*absence* d'un énoncé relié par une relation rhétorique de contradiction à l'énoncé qui contient le prédicat en question.

Calcul de l'ellipse sémantique

Pour un énoncé censé être produit par la machine, il peut être approprié d'omettre certains fragments, pour des raisons de lourdeur et de compréhension, c'est-à-dire, réaliser une *ellipse* de ces parties de l'énoncé. Cette omission peut être faite soit à un niveau purement linguistique [11], soit à un niveau sémantique, en supprimant certains prédicats dans la forme logique qui exprime la dénotation de l'énoncé que la machine est en train de produire [115]. Dans nos recherches, nous nous sommes intéressés au deuxième type d'ellipse, et pour la calculer nous nous basons sur la pile de prédicats (dans l'ontologie de tâches) qui expriment les connaissances des locuteurs en dialogue (à plusieurs locuteurs).

Ainsi, dans un contexte de dialogue à deux locuteurs, nous supposons que la pile de prédicats de l'utilisateur est mise à jour selon sa définition, donc l'algorithme pour contrôler l'ellipse précise que si l'énoncé censé être généré (par un des interlocuteurs, dont, éventuellement, la machine) contient un prédicat qui se trouve à la fois dans la pile et dans un énoncé antérieur relié à l'énoncé courant (censé être produit) *via* une relation rhétorique dialogique de confirmation, ou si le prédicat se trouve dans la pile, mais l'énoncé censé être produit est relié *via* une relation rhétorique monologique de contradiction à un énoncé qui, à son tour, est relié *via* une relation rhétorique dialogique de contradiction à un énoncé plus ancien, qui contient aussi le prédicat en question, alors nous supprimons ce prédicat dans l'énoncé courant ; sinon, si la relation rhétorique reliant l'énoncé censé être généré à un énoncé antérieur qui contient le prédicat, est de contradiction, ou le prédicat ne se trouve pas dans la pile, alors ce prédicat est conservé dans la forme logique de l'énoncé que la machine est en train de générer (voir le chapitre 6).

En ce qui concerne la mise à jour des piles de prédicats, nous y ajoutons tout prédicat qui est contenu (éventuellement sous forme niée) soit dans l'énoncé produit par l'utilisateur, soit dans un énoncé produit par la machine et confirmé (de manière explicite ou implicite) par l'interlocuteur (humain), *via* un énoncé qui est relié à l'énoncé produit par la machine *via* une relation rhétorique dialogique de confirmation ; à part ces éléments, nous n'ajoutons aucun autre prédicat dans la pile.

Il faut toutefois préciser que la réalisation *linguistique* de l'ellipse sémantique ne fait pas partie des recherches décrites dans cette thèse, mais nous attirons l'attention sur le fait qu'elle peut se manifester soit sous forme d'ellipse proprement dite, soit sous forme d'anaphore (par exemple, l'élimination d'un constituant GN – « groupe nominal » peut être marquée par une anaphore réalisée *via* le pronom 'en').

L'extension de ce mécanisme aux situations de dialogue à plusieurs locuteurs concerne deux aspects :

- la machine maintient un ensemble de piles de prédicats, une pile pour chaque interlocuteur ; ensuite, l'ensemble des prédicats censés être supprimés dans la forme logique représentant l'intention communicationnelle, censée être exprimée sous forme linguistique par la machine, est calculé en considérant l'*intersection* des piles de prédicats en tant qu'entrée dans la procédure de contrôle des ellipses ;
- la mise à jour des piles se fait en prenant en compte les **confirmations indirectes** des énoncés de la machine (voir ci-dessous).

Le premier aspect représente un choix de « bon sens », d'exprimer (sous forme linguistique) l'intention communicationnelle, de manière aussi explicite que nécessaire, afin qu'elle soit comprise par *tous* les destinataires du message contenu dans cette intention. Cela revient à ne supprimer, dans l'intention communicationnelle, que les prédicats qui sont « connus » par tous les destinataires (c'est-à-dire, les prédicats qui sont dans les piles de tous ces destinataires), tout en gardant les autres dans l'intention communicationnelle.

Le processus de mise à jour des piles de prédicats, présenté en détail dans le chapitre 6, s'appuie essentiellement sur le retour indirect fourni par les destinataires d'un énoncé, par conséquent, un prédicat contenu dans un énoncé est ajouté à la pile de prédicats d'un certain locuteur si :

- soit cet énoncé est produit par l'utilisateur en question ;
- soit cet énoncé est produit par la machine (ou par un autre locuteur, différent du locuteur concerné) et confirmé, de manière explicite ou implicite, par le locuteur concerné ; cette confirmation peut être soit **directe**, soit **indirecte**.

La confirmation indirecte est un cas particulier du retour indirect discuté ci-dessus (la situation où les deux relations rhétoriques impliquées sont du même type, donc, soit de contradiction, soit de confirmation) ; la confirmation directe est juste un énoncé qui est relié *via* une relation rhétorique dialogique de confirmation à l'énoncé produit par la machine. Dans le chapitre 6, ces types de confirmations (et d'infirmités⁶) sont présentés en détail.

Pour ce qui tient aux contraintes impliquées, le calcul de l'ellipse sémantique est guidé par des contraintes *dures passives* imposées par l'intersection de toutes les piles de prédicats des interlocuteurs humains, sur le contenu des énoncés ; ici, à partir de ces contraintes, une *recherche*

⁶Ici, par *infirmité* nous entendons le contraire, l'opposé de la confirmation.

est réalisée sur l'ensemble (totalement ordonné par rapport à l'inclusion) de toutes les formes logiques possibles qui expriment une intention communicationnelle ; ainsi, nous choisissons une forme sémantique qui satisfait les contraintes sur les piles de prédicats. De l'autre part, il y a un cas limite où ces contraintes deviennent *douces* : la situation où les piles de prédicats des usagers contiennent tous les prédicats dans l'énoncé censé être produit (par la machine) ; dans ce cas, la forme sémantique que la machine doit mettre sous forme linguistique serait l'ensemble vide. Par conséquent, le générateur de surface (réalisé *via* une « simple » manipulation d'expressions figées à trous dans notre cas) est instruit de produire un « *back-channel* » ou un énoncé de confirmation. Pour ce faire, nous remplaçons l'ensemble des prédicats dans la forme logique, avec un seul nouveau prédicat qui précise que l'énoncé courant est une confirmation (indéfinie), par rapport à tout ce qui avait été dit auparavant (pour produire une expression du type « uhum... », ou « OK... »).

Calcul du degré de force illocutoire

Notre perspective sur la force illocutoire qui peut être associée aux énoncés, ou, plutôt, qui est *réalisée* par les énoncés, prolonge des travaux antérieurs sur la formalisation de la Théorie des actes de langage [185] au sens où un énoncé est vu comme une fonction, le type d'acte de langage (un des suivants : F^F - « faire-faire », un *directif* dans la taxinomie de Searle [156] ; F^{FS} - « faire-faire-savoir », une requête d'informations ; F^S - « faire-savoir », un *assertif* selon Searle ; F^P - « faire-pouvoir », un acte déontique, offrant des choix au destinataire de cet acte ; F^D - « faire-devoir », un autre acte déontique, qui impose une obligation sur le destinataire de cet acte), qui est appliqué au contenu propositionnel (qui exprime, sous forme logique, la sémantique de l'énoncé). Le type d'acte de langage exprime plutôt la *connotation* de l'énoncé, tandis que le contenu propositionnel exprime sa *dénotation* [35] (voir chapitre 7).

De plus, pour une dénotation donnée (c'est-à-dire, un contenu propositionnel, exprimé sous forme logique) et une connotation (c'est-à-dire, un type d'acte de langage), l'énoncé proprement dit spécifié par ces éléments peut être réalisé avec un certain *degré de puissance* ; cet aspect est important notamment en génération des énoncés, car le succès du dialogue peut dépendre de manière cruciale du degré de puissance choisi pour réaliser sous forme linguistique une certaine intention communicationnelle. En accord avec des travaux antérieurs en dialogue homme-machine [35] et avec des recherches théoriques dans le domaine [185], nous avons choisi une échelle numérique de cinq valeurs, de -2 à $+2$, pour ce degré de puissance (voir chapitre 7).

Pour la génération des énoncés en dialogue à plusieurs locuteurs, il est important de concevoir un mécanisme pour choisir, de manière automatique, un certain degré de force illocutoire (dans l'ensemble des cinq valeurs précisées ci-dessus). Cela est motivé par le fait que, selon nous, le degré de force illocutoire devrait être choisi afin qu'un *arrière-plan commun* [39] soit acquis (atteint) aussi vite que possible par les partenaires du dialogue, et maintenu aussi longtemps que possible. D'un point de vue computationnel, cela revient au fait que plus les listes d'engagements des locuteurs ont des éléments en commun, plus près de 0 le degré de force illocutoire l'est (voir chapitre 7). Autrement dit, dans une situation idéale où les locuteurs partagent précisément les mêmes engagements tout au long du dialogue, la machine (c'est-à-dire, un des locuteurs, en fait) peut exprimer ses intentions de manière neutre (en attribuant donc un degré

de force illocutoire zéro à ses contributions dialogiques).

Cependant, dans les dialogues réels, pour la plupart des tours de parole, les listes d'engagements des locuteurs ne sont pas identiques ; les différences, en considérant des paires de locuteurs, y compris la machine et un locuteur humain⁷, peuvent essentiellement être de quatre types :

1. la liste d'engagements de la machine est incluse dans la liste d'engagements d'un locuteur humain ; ceci est typique pour des dialogues *via* lesquels l'utilisateur humain apprend des nouvelles tâches à la machine ; dans ce cas, la machine peut poursuivre deux stratégies :
 - (a) la machine accepte d'élargir sa liste d'engagements avec des faits pour lesquels l'utilisateur s'était engagé ; dans ce cas, l'énoncé produit par la machine a un degré faible de force illocutoire, -1 ;
 - (b) la machine essaye de convaincre l'utilisateur de renoncer à certains de ses engagements, afin que sa liste d'engagements soit identique à la liste de la machine ; dans ce cas, l'énoncé produit par la machine a un degré très fort de force illocutoire, $+2$;
2. la liste d'engagements de la machine inclut la liste d'engagements de l'utilisateur ; ceci est typique aux dialogues d'enseignements, où la machine apprend quelque chose à l'utilisateur ; dans ce cas, la machine peut poursuivre deux stratégies :
 - (a) la machine concède à rétracter certains énoncés antérieurs auxquels elle s'était engagée, et auxquels l'utilisateur ne s'était pas engagé ; dans ce cas, la machine choisit un degré très faible de force illocutoire, -2 ;
 - (b) la machine essaye de persuader l'utilisateur de s'engager pour d'autres énoncés (ou faits) auxquels la machine s'était engagée ; dans ce cas, la machine choisit un degré fort de force illocutoire, $+1$, pour l'énoncé qu'elle est en train de produire ;
3. il y a un certain chevauchement entre les listes d'engagements de la machine et de (des) utilisateur(s) ; ceci est typique pour les dialogues orientés service, où la machine interagit avec des utilisateurs *a priori* inconnus ; dans ce cas, la machine peut poursuivre quatre stratégies :
 - (a) la machine concède à enlever certains de ses faits dans ses engagements, afin que ses listes d'engagements contiennent seulement des faits auxquels l'utilisateur s'était, lui aussi, engagé, et essaye de convaincre l'utilisateur d'en faire autant ; dans ce cas, la machine produit deux énoncés, un avec un degré de force très faible (-2) et un autre avec un degré de force très fort ($+2$), respectivement ;
 - (b) la machine concède à s'engager pour d'autres énoncés supplémentaires, qui existent dans la liste d'engagements de l'allocutaire, tout en essayant de convaincre l'utilisateur de s'engager dans d'autres énoncés, existant dans la liste d'engagements de la machine ; dans ce cas, la machine génère deux énoncés, un avec un degré de force faible (-1), et un autre avec un degré de force fort ($+1$), respectivement ;

⁷Cette paire machine-humain est considérée plus attentivement dans nos recherches actuelles, car seule la génération des énoncés, c'est-à-dire, le moment où la machine est en train de produire une réplique, est le sujet de cette thèse.

- (c) la machine essaye de convaincre l'utilisateur de retirer quelques-uns de ses engagements (qui ne font pas partie également des engagements de la machine), et de s'engager dans d'autres énoncés (ou faits), auxquels la machine est engagée ; dans ce cas, la machine engendre deux énoncés, un avec un degré de force très fort (+2), et un deuxième, avec un degré de force fort (+1) ;
 - (d) la machine concède à retirer ses engagements, qui ne sont pas parmi les engagements de l'utilisateur, et à s'engager aux engagements de l'utilisateur, qui n'étaient pas les siens ; dans ce cas, la machine produit deux énoncés, un avec un degré de force très faible (-2), et un autre avec un degré de force faible (-1), respectivement ;
4. les listes d'engagements de la machine et de l'utilisateur n'ont rien en commun ; ceci est typique des dialogues échoués, où les locuteurs soit ont des arrière-plans culturels différents, soit ils se moquent l'un de l'autre ou du système ; cela est une version extrême du cas 3 ci-dessus, donc la machine peut adopter deux stratégies :
- (a) la machine renonce à tous ses engagements, pour accepter tous les engagements de l'allocutaire ; dans ce cas, un énoncé avec un degré de force très faible (-2) est produit par la machine ;
 - (b) la machine essaye de convaincre l'utilisateur de renoncer à tous ses engagements, pour ainsi s'engager à toute la liste d'engagements de la machine ; dans ce cas, un énoncé avec un très fort degré de force (+2) est généré par la machine.

La machine choisit la stratégie (parmi celles énumérées ci-dessus), en suivant l'heuristique selon laquelle l'ordinateur essaye de modifier aussi peu que possible sa propre liste d'engagements, en préférant y ajouter des faits, plutôt qu'en enlever.

En ce qui concerne les particularités du dialogue à plusieurs locuteurs, elles sont discutées en détail, pour ce qui tient au calcul du degré de force illocutoire, dans le chapitre 7. Essentiellement, le calcul du degré de force illocutoire des énoncés censés être produits par la machine est géré à l'instar du cas du dialogue à deux locuteurs, à l'exception de la mise à jour des listes d'engagements des locuteurs impliqués en conversation, et de la situation où un énoncé produit par la machine est destiné à *plusieurs* locuteurs (humains).

Essentiellement, les listes d'engagements des locuteurs impliqués en dialogue à plusieurs locuteurs sont mises à jour comme en dialogue à deux locuteurs, c'est-à-dire, de manière pilotée par les relations rhétoriques qui relient les énoncés, à l'exception de quelques détails : d'une part, il y a une différence entre la mise à jour des listes d'engagements des destinataires d'un énoncé généré par la machine, et la mise à jour des listes d'engagements des autres locuteurs. Plus précisément, les listes d'engagements des destinataires de l'énoncé de la machine sont mises à jour comme en dialogue à deux locuteurs, si ces destinataires ne produisent pas un **retour indirect** à l'énoncé de la machine (voir ci-dessous). Il n'est pas moins que les listes d'engagements des autres locuteurs sont mises à jour en fonction des énoncés que ces utilisateurs éventuellement produisent, c'est-à-dire, s'ils produisent des énoncés tels qu'il y a une relation rhétorique de confirmation entre chacun de ces énoncés et l'énoncé produit par la machine, alors les listes d'engagements sont mises à jour avec la sémantique de l'énoncé produit par la machine et, éventuellement, avec les sémantiques des énoncés produits par les utilisateurs ; sinon, s'il y a une relation rhétorique de contradiction entre chacun des énoncés des utilisateurs et l'énoncé

produit par la machine, alors les listes d'engagements sont mises à jour avec la sémantique niée de l'énoncé produit par la machine, et avec la sémantique de l'énoncé produit par chaque utilisateur, éventuellement avec la sémantique de la relation rhétorique qui relie ces deux énoncés ; si, pour chaque énoncé produit par ces locuteurs-là (qui ne sont pas les destinataires de l'énoncé produit par la machine), il n'y a aucune relation rhétorique reliant cet énoncé à l'énoncé produit par la machine, alors les listes d'engagements de ces locuteurs sont mises à jour en ajoutant seulement la sémantique de l'énoncé produit par chacun de ces locuteurs ; enfin, s'il n'y a aucun énoncé produit par ces locuteurs, alors leurs listes d'engagements restent non-modifiées.

En ce qui concerne le retour indirect que les destinataires de l'énoncé produit par la machine pourraient produire, cela revient à une réplique que ces interlocuteurs peuvent produire, réplique à un énoncé produit par d'autres locuteurs (qui ne sont pas destinataires de l'énoncé de la machine) ; cette réplique peut être reliée, *via* une relation rhétorique de contradiction ou de confirmation, à l'énoncé produit par la machine ; à son tour, le retour indirect peut être, lui aussi, relié à cette réplique. Si ces relations rhétoriques sont toutes les deux du même type (soit de confirmation, soit de contradiction), alors nous avons une contradiction indirecte par rapport à l'énoncé produit par la machine. Sinon (donc, s'il y a une confirmation indirecte), alors la liste d'engagements du locuteur en question est mise à jour avec l'énoncé produit par la machine ; sinon, cette liste est mise à jour avec la sémantique niée de l'énoncé produit par la machine et, éventuellement, avec la sémantique de la relation rhétorique qui relie cette réponse à l'énoncé produit par la machine.

Suite à cette discussion, nous pouvons voir que le calcul du degré de force illocutoire d'un énoncé revient à résoudre un problème de satisfaction de contraintes où des contraintes *actives* et *douces* sur le contenu (identique) des listes d'engagements des interlocuteurs, pour chaque ensemble de tours de dialogue où la machine est impliquée, pilotent un processus de *recherche* qui choisit le degré de force approprié, et cela d'une manière active, car la force est calculée suite à la satisfaction des ces contraintes.

Génération des connecteurs discursifs

Dans la plupart des dialogues il y a des tours de parole qui sont composés de plusieurs énoncés. Ici, par « énoncé » nous entendons la réalisation proprement dite d'une proposition (c'est-à-dire, une affirmation, interrogation ou assertion qui contient seulement un prédicat d'action). La cohésion de ces tours de parole à plusieurs énoncés est parfois assurée *via* des marques de ponctuation (virgule, point-virgule, points d'interrogation et d'exclamation, etc.) ; cependant, il est souvent besoin de mots particuliers entre ces énoncés, afin que le discours résultant soit cohérent et ait une *orientation argumentative* [119] appropriée. Dans ce cas, nous utilisons des mots tels que *mais*, *pourtant*, *bien que*, *quand même*, etc. Ainsi, quand plusieurs formes logiques (avec leurs types d'acte de langage correspondants) sont censées être générées dans le même tour de parole, il est intéressant et utile de pouvoir choisir les mots connecteurs appropriés entre ces formes logiques.

Ainsi, nous adaptons les sémantiques moeschleriennes de ces connecteurs, au sens où nous nous appuyons sur la structure rhétorique afin de dériver leur pertinence. Nous présentons de manière informelle ci-dessous, les contraintes (*dures*) qui pilotent la *recherche* dans l'ensemble des connecteurs possibles :

1. Les connecteurs *mais* et *quand même* sont définis *via* un « carré argumentatif » [118] délimité par trois énoncés, dont le troisième et sa forme niée sont les conclusions des premiers énoncés : ainsi, *mais* peut être utilisé de deux manières, selon le fait que le deuxième énoncé (c'est-à-dire, celui précédé par *mais*) est sous forme niée ou non ; dans le premier cas, *mais* peut être utilisé si les deux énoncés reliés par ce mot sont ainsi que le premier est un argument (ou prémisses) pour un autre (troisième) énoncé, le deuxième énoncé est un argument pour la forme niée de la conclusion du premier énoncé, et le deuxième énoncé est argumentativement plus fort que le premier ; de l'autre côté, la deuxième utilisation de *mais* entre deux énoncés est possible lorsqu'il existe un troisième énoncé qui est la conclusion du premier, ainsi que sa forme niée est la conclusion du deuxième énoncé. *Quand même* peut être utilisé entre deux énoncés lorsqu'il existe un troisième énoncé qui résulte *par défaut* du premier énoncé, et le deuxième énoncé est un argument pour la forme niée de ce troisième énoncé ;
2. Les connecteurs *pourtant* et *bien que* sont définis *via* un « triangle causal » [118], délimité par deux énoncés : *pourtant* peut être utilisé entre deux énoncés tels que le premier est la cause (logique) de la forme niée du deuxième ou inversement et, néanmoins, les deux énoncés sont tous les deux contextuellement vrais (c'est-à-dire, vrais dans un contexte (dialogique) particulier). *Bien que* peut être utilisé dans les mêmes conditions que *pourtant*, sauf qu'une contrainte supplémentaire doit être satisfaite : le temps de l'action décrite par l'énoncé qui précède ce connecteur doit précéder le temps de l'action décrite par l'autre énoncé.

Ces contraintes sont ensuite couplées avec la structure rhétorique (qui, en fait, les détermine) *via* un ensemble de correspondances entre les opérateurs de Moeschler (tels que « ou exclusif » – entre un énoncé et sa forme niée, la relation causale entre une prémisses et une conclusion, la relation de « norme » pour des implications qui sont vraies *par défaut*, et la relation argumentative, qui relie deux énoncés tels que le deuxième est un argument pour le premier [118]) et les relations rhétoriques. Ainsi, (i) le « ou exclusif » est mis en correspondance avec la relation rhétorique monologique *Contrast*, ou avec la relation dialogique de contradiction *P-Corr* (« Plan Correction »), (ii) la relation causale est mise en correspondance avec l'implication logique classique, (iii) la relation argumentative est calculée séparément (voir ci-dessous et chapitre 8) à partir de la structure rhétorique et des listes d'engagements des locuteurs, et (iv) la relation de « norme » est mise en correspondance avec la relation rhétorique monologique d'arrière-plan *Background*, ou avec la relation dialogique de question déterminante d'arrière-plan *Background_q*.

Dans ces contraintes, les énoncés qui ne sont pas reliés par ces mots concessifs, mais jouent un rôle dans la sémantique des connecteurs, sont récupérés soit du *contexte* discursif (c'est-à-dire, déduits à partir de la situation de dialogue, telle qu'elle est exprimée dans les listes d'engagements des interlocuteurs), ou du *cotexte* (c'est-à-dire, s'ils sont présents en tant qu'énoncés antérieurs dans le dialogue courant). Tous les mécanismes pour piloter le choix des connecteurs concessifs appropriés lors de la génération des tours de parole sont décrits dans le chapitre 8.

Un point essentiel pour inférer l'utilisation argumentative de *mais* consiste à déterminer la relation d'ordonnement argumentatif entre des paires d'énoncés ; il est donc important que, pour deux énoncés adjacents, nous puissions déterminer lequel d'entre eux est un argument plus

fort que l'autre, par rapport à une certaine conclusion (explicite ou implicite). Ainsi, la relation d'ordre argumentatif ne dépend pas seulement des énoncés qui sont comparés, mais aussi des conclusions *potentielles* que ceux-ci peuvent déterminer. Afin d'accomplir ce *desideratum*, nous avons formalisé, dans une logique du premier ordre, une partie plus récente de la Théorie de l'argumentation de Anscombe et Ducrot [6], [58]. Le mécanisme conçu consiste essentiellement à prendre deux énoncés en entrée et décider si ces deux énoncés peuvent être comparés (c'est-à-dire, s'il existe un énoncé tel que le premier énoncé en est un argument, et le deuxième énoncé est aussi un argument pour cet énoncé, ou pour sa forme niée), ou non et, si oui, préciser lequel des deux énoncés initiaux est argumentativement plus fort. Ainsi, l'évaluation argumentative des (paires d')énoncés comprend les étapes suivantes :

1. pour n'importe quelle paire d'énoncés, nous cherchons les topoï non-disjoints qui ont ces énoncés comme prémisses (voir le chapitre 8) ; cela revient à chercher des paires telles que 'Plus / moins on fait quelque chose, plus / moins on obtient quelque chose' ; si cela est vrai, alors nous continuons avec l'étape suivante ; sinon, nous décidons que les deux énoncés ne sont pas comparables du point de vue argumentatif ;
2. on choisit la première paire rencontrée de topoï non-disjoints déterminés à l'étape précédente : si les deux topoï sont soit intrinsèques (c'est-à-dire, leur valeur de vérité est déterminée par des connaissances par défaut) ou extrinsèques (c'est-à-dire, leur valeur de vérité est déterminée à partir des connaissances factuelles reliées au dialogue), alors nous regardons davantage (le cas échéant) le type des topoï intrinsèques dont il s'agit et, par conséquent, nous situons les deux énoncés par rapport à leur puissance argumentative (voir chapitre 8 pour une présentation détaillée de ce mécanisme).

Du point de vue de la satisfaction des contraintes, la structure rhétorique impose des contraintes *dures* sur la *sélection* du connecteur ; il n'est pas moins qu'afin de situer les énoncés du point de vue argumentatif, nous avons besoin d'informations *a posteriori*, concernant les conclusions que les énoncés déterminent (ou peuvent déterminer) ; par conséquent, une approche purement à base de contraintes où seulement des informations *a priori* disponibles soient utilisées, n'est pas applicable directement.

1.3.4 Efficacité des démarches proposées : illustration

Dans ce paragraphe nous allons considérer de manière informelle quelques exemples de dialogues homme-machine à plusieurs locuteurs, pour illustrer l'effet du cadre proposé dans cette thèse, sur les énoncés que la machine produit en dialogue. Plus précisément, nous allons nous concentrer sur les effets que notre cadre a sur les formes de surface pour les énoncés de la machine. Ainsi, considérons par exemple un dialogue entre une machine (qui joue le rôle d'une bibliothécaire virtuelle) et deux locuteurs humains, autour d'une tâche de réservation de livres dans une bibliothèque⁸. Tout d'abord nous présenterons le dialogue où les répliques de la machine sont calculées en branchant directement la sortie du contrôleur de dialogue à l'entrée dans un générateur de surface (à base d'expressions figées) ; ensuite, nous présenterons le dialogue

⁸Cet exemple sera repris dans le chapitre 8 pour une discussion plus approfondie de certains aspects concernant le choix des connecteurs discursifs.

obtenu en appliquant les démarches proposées dans cette thèse. Nous notons par $U^{(1)}$ et $U^{(2)}$ les locuteurs humains (les clients de la bibliothèque), et par M , la machine.

Ainsi, si nous ne faisons qu'une mise en correspondance entre les formules logiques qui expriment les intentions communicationnelles calculées par le contrôleur de dialogue, et des fragments d'énoncés en langue naturelle, les répliques de la machine se présenteraient comme ci-dessous (les indices inférieurs indiquent la succession des tours de parole) :

$U_1^{(1)}$: Je voudrais emprunter une tragédie ancienne.

$U_1^{(2)}$: Moi aussi, de Sophocle.

M_1 (en parlant aux *deux* clients) : Je ne peux pas vous prêter une tragédie ancienne de Sophocle ou tout autre livre dans notre bibliothèque, vous avez emprunté cinq livres dans notre bibliothèque.

$U_2^{(1)}$: Oui, mais nous avons un examen demain, pourriez-vous faire une exception ?

M_2 (en parlant à $U^{(1)}$) : Je pourrais vous prêter une tragédie ancienne de Sophocle, vous n'avez jamais eu de retard dans la restitution des livres à la bibliothèque.

M_2 (en parlant à $U^{(2)}$) : Je ne pourrais pas vous prêter une tragédie ancienne de Sophocle, il n'existe plus qu'un exemplaire de tragédie ancienne de Sophocle dans la bibliothèque, vous n'avez pas retourné à la bibliothèque les livres que vous avez empruntés à la bibliothèque le 13 mai 2007.

$U_2^{(2)}$: Donc, pourrais-je lire le livre dans la bibliothèque, en consultation sur place ?

Nous observons que par cette simple approche la machine produit des énoncés lourds et peu naturels, où les répétitions abondent, les anaphores sont presque absentes (à l'exception des pronoms personnels qui désignent les locuteurs – ceux-ci sont facilement « calculés » en mettant toujours en correspondance le destinataire d'un énoncé avec le pronom personnel 'vous') et les connecteurs discursifs manquent complètement.

Certains de ces problèmes pourraient être résolus strictement au niveau du générateur linguistique de surface, *via* des règles qui spécifient des faits tels que : une deuxième apparition d'un couple (objet, déterminant) dans un même énoncé peut entraîner la suppression du déterminant dans cette deuxième apparition. Ces démarches amèneraient la machine à avoir un comportement comme nous le montrons ci-dessous :

$U_1^{(1)}$: Je voudrais emprunter une tragédie ancienne.

$U_1^{(2)}$: Moi aussi, de Sophocle.

M'_1 (en parlant aux *deux* clients) : Je ne peux pas vous prêter une tragédie ancienne de Sophocle ou tout autre livre dans notre bibliothèque, vous avez emprunté cinq livres.

$U_2^{(1)}$: Oui, mais nous avons un examen demain, pourriez-vous faire une exception ?

M'_2 (en parlant à $U^{(1)}$) : Je pourrais vous prêter une tragédie, vous n'avez jamais eu de retard dans la restitution des livres à la bibliothèque.

M'_2 (en parlant à $U^{(2)}$) : Je ne pourrais pas vous prêter une tragédie, il n'existe plus qu'un exemplaire de tragédie, vous n'avez pas retourné à la bibliothèque les livres que vous avez empruntés le 13 mai 2007.

$U_2^{(2)}$: Donc, pourrais-je lire le livre dans la bibliothèque, en consultation sur place ?

Nous remarquons que dans ce cas également, les énoncés de la machine ne sont pas trop naturels et, de plus, les connecteurs discursifs ne sont toujours pas présents, ce qui ne nous permet pas de calculer facilement les relations logiques entre les diverses énoncés dans un même tour de parole.

En revanche, si le passage de l'intention communicationnelle vers la forme de surface est contrôlé par les mécanismes annoncés dans ce chapitre et décrits *in extenso* dans cette thèse, alors la machine sera amenée à produire des répliques plus naturelles et cohérentes :

$U_1^{(1)}$: Je voudrais emprunter une tragédie ancienne.

$U_1^{(2)}$: Moi aussi, de Sophocle.

M_1'' (en parlant aux *deux* clients) : Je ne peux vous prêter plus rien, parce que vous avez déjà emprunté cinq livres...

$U_2^{(1)}$: Oui, mais nous avons un examen demain, pourriez-vous faire une exception ?

M_2'' (en parlant à $U^{(1)}$) : Pour vous, oui, car vous êtes un bon client

M_2'' (en parlant à $U^{(2)}$) : mais pas pour vous, parce que je n'ai plus qu'un exemplaire dans la bibliothèque, et vous n'avez pas encore retourné les livres que vous avez empruntés il y a cinq mois !

$U_2^{(2)}$: Donc, pourrais-je lire le livre dans la bibliothèque, en consultation sur place ?

1.4 Discussion

Dans ce chapitre nous avons présenté, de manière résumée, une approche à base de contraintes, sur des bases pragmatiques, pour le contrôle de la génération des énoncés par la machine, en dialogue à plusieurs locuteurs. Cette approche constitue un cadre rigoureux et implémentable dans des applications de dialogue générique⁹, orienté service. Après un bref aperçu des bases des problèmes de satisfaction de contraintes, nous avons montré comment ces problèmes s'instancient pour guider la génération des énoncés.

Un des aspects les plus importants de notre cadre, par rapport aux approches plus « traditionnelles », dans la parenté de l'approche BDI, est le fait de ne reposer que sur les engagements ouvertement assumés par les locuteurs, et sur la structure rhétorique résultant de l'enchaînement de leurs contributions. Ainsi, nos travaux, d'un côté, prolongent, en formalisant, des travaux antérieurs de Hamblin [73] et, d'un autre côté, se situent en complémentarité avec des recherches telles que celles de Kibble [91], où, à partir d'une taxinomie plus raffinée des engagements publics – tributaire de Brandom [24], on propose, dans un premier temps, un cadre pour guider la planification des tours de parole à partir de ces engagements [91] pour ensuite relier ces engagements aux représentations rhétoriques, dans le cadre de la RST de Mann and Thomson [92].

En conclusion, nos contributions principales portent sur :

- la proposition d'un cadre à base de contraintes pour piloter les décisions en génération des énoncés ;

⁹Tout au long de cette thèse, par « généralité » nous entendons une séparation nette entre les aspects indépendants de la tâche particulière considérée par l'application, et les aspects dépendants de cette tâche.

- la conception d'un composant de structuration rhétorique des répliques dialogiques ; pour ce faire, nous avons pris comme élément de départ la théorie SDRT et une taxinomie des types d'actes de langage qui caractérisent les visées illocutoires des énoncés ; ensuite, nous avons proposé un ensemble de sémantiques dans une logique du premier ordre, pour certaines relations rhétoriques de la SDRT ; ces sémantiques subissent ensuite des contraintes contextuelles, que nous avons formalisées à partir de la notion de grammaire abstraite d'attributs, utilisée dans la vérification des programmes informatiques ;
- le couplage entre l'aspect structurel concernant la représentation rhétorique des énoncés en dialogue, et des phénomènes linguistiques de surface, notamment le contrôle de l'ellipse sémantique, le calcul du degré de force illocutoire, et le choix des connecteurs discursifs ; ainsi, pour les ellipses sémantiques, nous nous avons modélisé les connaissances ou faits auxquels les interlocuteurs se sont engagés, pour ainsi utiliser la structure rhétorique afin de décider sur la pertinence des constructions elliptiques ; pour le degré de force illocutoire, nous avons couplé la structure rhétorique avec les engagements publiquement assumés par les locuteurs, en posant comme critère de succès pour le dialogue l'équivalence sémantique entre les engagements des interlocuteurs ; pour la génération des connecteurs, nous avons proposé un ensemble de contraintes rhétoriques sur leur choix, en passant également par des considérations concernant les valences argumentatives des énoncés. ;
- l'extension de tout le cadre proposé au dialogue à plusieurs locuteurs ; pour ce faire, nous avons considéré ce type de dialogue comme résultant de certaines superpositions de dialogues à deux locuteurs, en formalisant ensuite les contraintes induites par la simultanéité de ces dialogues.

Toutes ces contributions ont été validées sur des exemples typiques de dialogues, autour d'une tâche de réservation de livres dans une bibliothèque publique.

Chapitre 2

Etat de l'art : systèmes de génération face aux défis du dialogue

2.1 Génération du langage naturel pour le dialogue homme-machine¹

2.1.1 Contexte du problème

Il est généralement accepté que la génération du langage naturel signifie la création du langage naturel, de façon automatique et à partir d'une représentation non-linguistique de l'information à communiquer. En tant que représentations non-linguistiques on peut avoir :

- des informations extraites d'une base de données ;
- une spécification abstraite d'un message, produite par le contrôleur de dialogue, dans un système de dialogue homme-machine.

La plupart des travaux en génération de la langue concernent la production des textes écrits. Il est généralement accepté que ce processus consiste en trois étapes [147] :

1. Planification du document :

- *détermination du contenu* – cette tâche suppose la conversion de l'entrée du système vers un ensemble d'objets spécifiques au système de génération. Ces objets constituent la base des traitements ultérieurs dans le système de génération. Un problème important à ce niveau est le choix des informations à exprimer dans le texte qui sera généré.
- *structuration du contenu* (ou planification du discours) – cette tâche suppose l'ordonnement des informations qui devront être exprimées et l'établissement d'une structure pour le texte à générer. Le résultat est constitué d'un plan de texte, souvent sous forme de structure arborescente représentant l'ordonnement et le regroupement des messages, ainsi que les relations entre eux.

2. Microplanification :

- *lexicalisation* – cette tâche suppose le choix des mots appropriés pour l'expression des informations en entrée. Souvent un concept peut être exprimé en employant des mots ou des expressions différents ; la tâche de la lexicalisation revient au choix des mots et des expressions les plus appropriés dans le contexte donné.

¹Cette section est basée en grand partie sur un rapport de Theune [170].

- *agrégation* – cette tâche suppose la décision sur les informations à exprimer dans une phrase. Des « morceaux » d'information constituant des messages d'entrée différents peuvent être joints et exprimés dans une seule phrase qui pourrait contenir, par exemple, une conjonction ou une expression relative.
- *génération des références* – cette tâche suppose la création des expressions identifiant les entités dans le domaine d'utilisation pour le système. Cela implique le choix du type d'expression (par exemple, un pronom ou une description définie) et, dans le cas d'une description définie, le choix des propriétés à inclure dans la description.

3. Réalisation de surface :

- *réalisation linguistique* – cette tâche suppose la création des phrases grammaticales. Cela implique l'application des règles syntaxiques et morphologiques afin de contrôler des aspects tels que l'accord grammatical.
- *réalisation structurelle* – cette tâche suppose la conversion du texte de sortie dans un format externe requis, par exemple en ajoutant des balises HTML ou LaTeX. Cela ne constitue pas une tâche de génération en soi et dans beaucoup de systèmes elle n'est pas traitée de manière explicite.

En général, nous pouvons dire que les tâches de planification du document de l'étape 1 sont indépendantes de la langue et spécifiques au domaine d'utilisation, tandis que les tâches de réalisation superficielle de l'étape 3 sont spécifiques à la langue mais, en principe, indépendantes du domaine d'utilisation.

Les tâches associées à l'étape 2, de microplanification, demandent à la fois des connaissances dépendantes du domaine et des éléments spécifiques à la langue d'utilisation. Par exemple, la génération des références nécessite des connaissances sur le domaine (informations sur les objets du domaine et sur leurs propriétés) et sur le langage (les propriétés syntaxiques des modificateurs exprimant certaines propriétés des objets du domaine).

La plupart des systèmes de génération ont pour but la production de « morceaux » de textes, sans intervention humaine. La génération du langage naturel est très rarement utilisée pour la génération des énoncés du système dans le cadre des dialogues avec un partenaire humain. Quelques raisons pour cet état des choses sont explicitées ci-dessous, où nous discutons les différences entre la génération du langage dans une situation monologique par rapport à une situation dialogique.

2.1.2 Génération monologique *versus* génération dialogique

Dans un système de dialogue, l'entrée dans le composant de génération langagière (si cela existe) consiste, en général, dans une spécification de message, produite par le contrôleur de dialogue. Cette spécification de message décrit le contenu à exprimer dans le tour de parole suivant du système. Ceci signifie que c'est le contrôleur de dialogue qui prend les décisions sur le « quoi dire » et sur le « quand le dire », réalisant ainsi les tâches de génération reliées à la détermination du contenu et, *en partie*, à la planification du discours.

En général la responsabilité du composant de génération consiste à *formuler* les énoncés du système, c'est à dire, à décider sur le « comment le dire ». Ce processus est parfois dénommé

« génération tactique » [147] et comprend en principe toutes les tâches correspondantes aux étapes de microplanification et de réalisation superficielle dans le processus de génération.

Beaucoup de systèmes de génération fonctionnent dans un domaine d'application si restreint qu'il est possible de ne réaliser les tâches de microplanification que de façon très limitée ou même pas du tout. Cela est vrai en particulier pour la plupart des applications dialogiques, telles que les renseignements sur le trafic aérien [117]. Souvent, les énoncés à être produits par ce type de systèmes de dialogue contiennent une seule phrase (soit une question, soit une réponse courte), donc la réalisation de l'agrégation est à peine nécessaire. De plus, la lexicalisation et la génération des références requiert en général des connaissances linguistiques très limitées, puisque la plupart des systèmes fonctionnent dans des domaines à restrictions, où les expressions stéréotypiques sont communes. Cela revient au fait que seule la réalisation linguistique reste la tâche essentielle pour la génération des énoncés en dialogue. Cette tâche peut souvent être réalisée *via* une simple manipulation des suites de caractères, car le domaine de dialogue est d'ordinaire tellement restreint qu'il est possible de préciser *a priori* toutes les chaînes textuelles de sortie (suite, bien sûr, à l'abstraction préalable sur les informations variables, telles que des coordonnées temporelles ou spatiales précises par exemple).

Bref, dans beaucoup de systèmes pratiques de dialogue un composant complet de génération linguistique n'est apparemment pas nécessaire, car des techniques relativement simples suffisent pour produire des énoncés linguistiques adéquats. Néanmoins, la génération des énoncés *via* de telles approches simplistes fait preuve d'un important manque de flexibilité et de sensibilité au contexte. Cela pourrait constituer un vrai problème dans des domaines ayant besoin d'un dialogue plus sophistiqué où un comportement figé est moins acceptable, comme par exemple les dialogues d'enseignement ou les dialogues sociaux de type e-démocratie (y compris ceux qui ont pour but le commerce électronique – le e-commerce).

Les locuteurs humains en dialogue prennent en compte plusieurs aspects différents du contexte dialogique lors de la formulation de leurs réponses. Par exemple, les informations sur la forme et le contenu des énoncés antérieurs dans le dialogue sont utilisées pour produire des énoncés elliptiques et des anaphores pronominales des entités mentionnées antérieurement. De plus, le style des énoncés dialogiques dépend de la relation entre le locuteur et l'allocutaire (par exemple, professeur-étudiant, institution-client, ou administration-citoyen), des différentes caractéristiques de l'allocutaire (novice ou expert, adulte ou enfant) et de l'état émotionnel et cognitif de l'allocutaire (heureux ou malheureux, sûr ou pas). Théoriquement, un système pour la génération automatique des énoncés dialogiques devrait prendre en compte tous ces aspects contextuels. Evidemment, la réalisation de cette démarche *via* une simple manipulation des suites de caractères serait extrêmement difficile, sinon impossible, donc, on a besoin d'une forme plus avancée de génération linguistique. Dans le paragraphe concernant l'état de l'art sur la génération dialogique nous verrons plusieurs approches différentes, illustrées par des systèmes pratiques existants.

2.2 Approches en génération du langage

2.2.1 Position du problème

Comme nous avons déjà vu², la spécification du message produite par le contrôleur de dialogue dans un système de dialogue homme-machine contient des informations concernant l'acte de langage, les prédicats et les arguments du message à générer dans la contribution dialogique suivante. Pour pouvoir produire un énoncé en langage naturel à partir d'une telle spécification de message on a besoin d'une forme de génération de langage.

Dans cette section nous présentons quatre approches possibles pour la génération des énoncés dialogiques, correspondant à quatre types de systèmes de génération de langage qui pourraient être utilisés pour notre système de dialogue. Pour cela, nous présenterons de manière concise les avantages et les inconvénients de chaque approche.

2.2.2 Générateurs à base d'analyse inverse

Dans l'analyse du langage la forme de surface d'une expression en langage naturel est associée à une représentation de son sens, *via* l'utilisation d'une « grammaire » qui précise la relation entre le sens et la forme de surface. La génération du langage (au moins dans le contexte des systèmes de dialogue) commence avec une représentation du sens, et associe celle-ci à une forme de surface, afin que ce sens soit exprimé. Du fait que l'analyse et la génération utilisent la relation entre sens et forme de surface, mais dans des sens opposés, la considération de la génération en tant qu'analyse inverse est une idée naturelle ; ainsi, on est tenté d'utiliser une seule grammaire à la fois pour l'analyse et pour la génération. Les avantages importants de cette approche d'« analyse inverse » sont :

- l'utilisation du même formalisme à la fois pour l'analyse et pour la génération ;
- l'utilisation d'un seul format pour la représentation du sens, dans tous les composants du système : la sortie de l'analyse a le même format que l'entrée dans le composant de génération ; ainsi, l'entrée est consistante avec la sortie (du point de vue du contrôleur de dialogue – [35], mais cf. aussi chapitre 1 de cette thèse) – le système peut analyser tout ce qu'il peut générer et inversement.

Malgré la simplicité séduisante de cette idée, l'approche d'« analyse inverse » a été à peine considérée sérieusement lors de l'implémentation pratique des systèmes de génération. Une raison pour cela est qu'en pratique, l'analyse et la génération utilisent si différemment une grammaire « réversible » qu'il est très difficile à utiliser la même grammaire pour les deux buts. Comme on montre dans [147], un module d'analyse pourrait produire la même formule logique pour plusieurs énoncés textuels différents exprimant essentiellement le même sens (propositionnel). Ainsi, l'analyse inverse ne peut pas réaliser les actions correspondantes aux étapes de microplanification (choix des lexèmes, choix des expressions référentielles, agrégation) ; cette approche ne peut accomplir que l'étape de réalisation linguistique de surface.

Une autre raison, d'importance relativement moindre, réside dans le fait que les grammaires pour l'analyse sont conçues, en général, de manière robuste, afin qu'elles puissent manier convenablement les entrées fragmentées ou au moins agrammaticales, tandis qu'une grammaire pour

²Ce paragraphe est basé dans une mesure importante sur un rapport de Theune [170].

la génération devrait, évidemment, produire seulement des sorties correctes et complètes du point de vue grammatical. Egalement, les grammaires d'analyse intègrent souvent des informations utiles pour résoudre les ambiguïtés, ce qui n'est pas nécessaire pour la génération.

Plusieurs des problèmes précisés ci-dessus pourraient être résolus *via* des grammaires conçues de manière *ad-hoc*, adéquates pour la génération, au lieu d'essayer d'employer la même grammaire pour la compréhension et pour la génération. Une telle grammaire orientée vers la génération pourrait rendre compte seulement des expressions que le système devrait être capable de générer. Cette dernière approche est suivie dans la plupart des systèmes de génération basés sur une grammaire [170]. C'est pour cette raison que nous utiliserons désormais le syntagme « analyse inverse » pour désigner les systèmes de génération basés sur une grammaire, qui prennent en entrée une représentation logique du sens, semblable aux représentations produites par les grammaires d'analyse – c'est à dire, une entrée qui n'a pas *a priori* pour but la génération (il s'agit, donc, d'une représentation différente des plans syntaxiques détaillés requis par la plupart des générateurs de surface spécialisés, comme nous le verrons dans le paragraphe suivant).

L'« analyse inverse » (dans le sens élargi précisé ci-dessus) pourrait être une approche faisable de la génération du langage dans des systèmes simples de dialogue, où l'ensemble des énoncés à produire est très limité. Par exemple, beaucoup de ces systèmes ont besoin de produire seulement des phrases à la voix active, à l'indicatif présent ; cela rend inutile le choix concernant le temps ou la voix des énoncés. De plus, le vocabulaire utilisé en génération pourrait être très limité, ainsi à chaque concept on peut associer un et seulement un mot ou syntagme ; cela revient à annuler le problème des choix lexicaux (de synonymes). Enfin, dans ces systèmes, l'entrée à la génération peut être offerte dans des segments qui correspondent déjà à des phrases, ce qui rend l'agrégation inutile.

Un inconvénient important de cette approche réside dans le manque de choix, ce qui revient au fait que la sortie du système ne peut pas être adaptée aux contextes de sa production : la même représentation du contenu sera toujours associée à la même représentation de surface. Par exemple, le générateur ne peut décider d'employer un pronom après avoir déjà fait référence au même objet lors des tours de parole antérieurs si cela ne lui est précisé de façon explicite en entrée – c'est-à-dire, ce type de décisions est plutôt de la responsabilité du contrôleur de dialogue que du composant de génération.

Bref, il apparaît que l'approche d'« analyse inverse » n'offre pas beaucoup plus de flexibilité par rapport à la simple manipulation des suites de caractères, bien qu'elle soit une solution théoriquement beaucoup plus élégante au problème de la génération.

2.2.3 Générateurs textuels à base de grammaires

Le syntagme « générateur textuel » désigne ici précisément un système informatique particulièrement conçu pour la tâche de réalisation linguistique, c'est à dire, pour déterminer la forme de surface d'un énoncé. Un générateur textuel est conçu précisément pour résoudre le problème du choix de la forme de surface la plus appropriée, s'il y en a plusieurs disponibles. La plupart des générateurs textuels connus sont capables de générer des phrases seules, qui pourraient ou non faire partie d'un texte plus étendu. Les générateurs textuels prennent en entrée une certaine spécification pour la phrase à générer, ou « plan de la phrase », qui est beaucoup plus riche

qu'une représentation logique standard du contenu sémantique de l'énoncé. Afin d'associer cette entrée à une phrase grammaticale il y a deux possibilités : (i) l'utilisation d'une grammaire spécialisée ou (ii) l'utilisation des structures phrasales figées (en anglais, « sentence templates »), où des parties de l'énoncé de sortie sont *a priori* précisées.

Les générateurs textuels basés sur une grammaire offrent une méthode élégante du point de vue théorique pour la génération des énoncés dialogiques. La génération est basée sur l'emploi d'une grammaire, théoriquement motivée et de large couverture, pour le langage cible dans la génération. Le fait que la grammaire est adéquate à la génération signifie qu'on évite tous les problèmes concernant les grammaires bidirectionnelles, discutés dans la section précédente. Si le générateur textuel possède une grammaire pour la langue cible du système, alors le coût d'adéquation du générateur à une application quelconque est assez bas.

Evidemment, l'utilisation d'un générateur textuel basé sur une grammaire dans le cadre des systèmes de dialogue a des inconvénients de nature pratique surtout. Tout d'abord, l'entrée dans les générateurs textuels actuels est beaucoup plus riche et plus précise que les représentations logiques produites d'ordinaire par les contrôleurs de dialogue. Une telle représentation riche pour l'entrée dans un générateur textuel peut contenir, entre autres, la spécification des lexèmes à utiliser lors de la description des divers concepts. Afin d'amener une représentation logique fournie par le contrôleur de dialogue à la richesse requise par un générateur textuel on a besoin d'une composante supplémentaire de microplanification. Lors de la planification, en partie spécifique au domaine de l'application, les informations concernant les lexèmes et leurs formes fléchies – temps, nombre, genre, etc. – sont ajoutées à la spécification logique du message.

Pour la plupart des systèmes de dialogue existants à l'heure actuelle, l'utilisation d'une grammaire de large couverture serait trop coûteuse. Bien que telles grammaires sont théoriquement intéressantes, leur taille et leur complexité représentent souvent des problèmes d'ordre pratique. Par exemple, les générateurs textuels (linguistiques) peuvent être assez lents, surtout lors de la génération des énoncés relativement longs. Evidemment, ceci ne convient pas du tout aux dialogues oraux spontanés, par exemple.

2.2.4 Générateurs textuels à base de structures figées

Une méthode alternative de réalisation linguistique superficielle, beaucoup plus acceptable du point de vue pratique, est l'utilisation des expressions linguistiques figées. Une expression linguistique figée est « une forme prédéfinie qui est alimentée avec des informations, soit par l'utilisateur, soit par l'application, lors du fonctionnement » [170]. Cette définition est plutôt large, car, comme nous le verrons plus loin dans ce chapitre, les structures figées utilisées par les systèmes de génération textuelle sont de natures assez différentes.

Bien que théoriquement moins élégante que la génération textuelle basée sur une grammaire, l'utilisation des structures figées a un avantage pratique important, la vitesse de génération, car la grandeur et le nombre des structures et règles à essayer et bien inférieur à celles d'une grammaire. En général, le développement d'une nouvelle application est plus facile *via* l'utilisation des expressions figées que *via* une grammaire, car la spécification des structures figées nécessite moins d'expertise linguistique, que l'ajout ou l'adaptation des règles dans une grammaire. Un autre avantage des systèmes basés sur des structures figées est que ceux-ci peuvent gérer des

entrées moins détaillées du point de vue linguistique, par rapport aux entrées attendues par une grammaire.

Du fait que les structures figées sont construites à la main ou collectées dans des textes ou dans des ressources linguistiques comme l'Internet, des structures figées différentes peuvent être spécifiées pour des contextes différents. De même que les générateurs employant une grammaire, les générateurs à base de structures figées attendent que toute l'information contextuelle soit offerte par une composante de planification.

Les systèmes basés sur des structures figées peuvent donc être utilisés pour la génération du langage dans un éventail très large d'applications ; cependant, ils sont plus appropriés pour certaines applications que pour d'autres, par exemple pour l'utilisation dans des domaines de taille relativement réduite dont la structure et l'organisation changent très peu ou pas du tout. Enfin, un autre inconvénient des générateurs linguistiques à base de structures figées réside dans la forte dépendance du domaine et de l'application ; lors de l'utilisation dans une application différente, à peu près tout l'ensemble de structures figées doit être recréé.

2.2.5 Utilisation des systèmes complets de génération

La tâche principale d'un composant de génération dans le contexte d'un système de dialogue homme-machine est la réalisation superficielle d'un énoncé-réponse. Nous avons vu (cf. chapitre 1) qu'au moins dans des dialogues sophistiqués l'accomplissement de plusieurs tâches concernant la microplanification – choix lexical, génération des références – est nécessaire. Lors de l'utilisation d'un système complet de génération, toutes ces tâches sont, en principe, accomplies. Un système complet de génération peut prendre en entrée des données non linguistiques ; en général, n'importe quel type d'entrée peut être choisi, autant qu'il a une structure prédictible (cohérente). Un autre avantage d'une telle approche est qu'un système de génération complet peut être utilisé à la fois pour la génération des interventions courtes dans le dialogue, et des énoncés plus longs et élaborés. Cette dernière tâche ne peut pas être satisfaite par un simple générateur textuel, car il manque de capacités de planification.

Un inconvénient important de l'utilisation des systèmes de génération complets pour le dialogue est que la génération dialogique n'a pas besoin de toute la « puissance » d'un tel système de génération (complet). Ainsi, la génération des contributions dialogiques n'implique pas, à vrai dire, la planification du document (cette tâche est accomplie par le contrôleur de dialogue) et le rôle de la microplanification est, lui aussi, assez limité. Cela signifie qu'un système de génération complet est sur-équipé pour la tâche et que certaines composantes de sa structure devront être supprimées ou au moins modifiées. Par exemple, on doit supprimer la planification du document, en gardant en même temps la possibilité de gérer les entrées non-linguistiques, d'une manière fragmentée par la succession des tours de parole. De plus, étant donné qu'un système complet utilise des informations spécifiques au domaine (au niveau de la planification du document et de la microplanification), l'adaptation d'un tel système à une nouvelle application peut être relativement coûteuse.

2.3 Systèmes de génération du langage, face aux défis du dialogue

2.3.1 Position du problème

Comme nous venons de le préciser, beaucoup de systèmes de dialogue en langage naturel utilisent du « texte figé » pour la génération de leurs réponses [117], [116], [146], [165]. Une telle approche pourrait suffire pour des dialogues dans des domaines restreints où les énoncés produits par le système sont courts, simples, et utilisent des expressions figées. Néanmoins, pour les dialogues plus sophistiqués on a besoin d'une méthode plus avancée pour générer les contributions du système.

Tout d'abord, un générateur pour le dialogue devrait être sensible au contexte, c'est-à-dire qu'il faudrait prendre en compte à la fois l'historique du dialogue et sa cohérence (souhaitée). Ensuite, l'utilisateur devrait être pris en compte, au moins en ce qui concerne son niveau de familiarité au thème du dialogue et la relation interactive avec la machine. De plus, lorsqu'il s'agit de dialogues oraux, le système doit fournir des éléments qui puissent rendre expressifs les énoncés à produire : par exemple, le système doit pouvoir gérer la prosodie, ou au moins l'accentuation des énoncés.

2.3.2 Générateurs à base d'analyse inverse

Les systèmes que nous allons présenter de manière succincte ici sont ALE et ASTROGEN ; ils prennent en entrée des structures pas particulièrement conçues pour la génération. Cependant, ces systèmes n'utilisent pas de véritables grammaires bidirectionnelles, mais des grammaires spécialement adaptées à la génération. ASTROGEN est un petit système conçu particulièrement pour la génération du langage naturel. ALE, à son tour, est un outil-cadre générique pour le traitement du langage naturel, permettant de faire à la fois l'analyse et la génération du langage. Le système GEMINI, le troisième dans cette catégorie, utilise à vrai dire une grammaire bidirectionnelle pour l'analyse et pour la génération. GEMINI prend en entrée pour la génération une formule logique enrichie au préalable avec des informations pertinentes pour la génération.

Le système ALE

ALE (« Attribute Logic Engine ») est un système intégré pour l'analyse et la génération du langage naturel [170]. Ce système, basé sur les grammaires HPSG (« Head-driven Phrase Structure Grammar ») et LFG (« Lexical Functional Grammar »), contient un environnement intégré pour créer des grammaires, un analyseur linguistique et un générateur.

En tant qu'entrée pour la génération, ALE prend des informations sémantiques minimales, accompagnées par une catégorie de buts syntaxiques. Cependant, un problème pour ce système consiste dans le choix de l'énoncé textuel le plus approprié parmi les énoncés candidats (voire paraphrases). Ce système est implémenté en Prolog.

Le système ASTROGEN

Le système ASTROGEN (« Aggregated Deep and Surface Natural Language Generator ») est un générateur du langage naturel [170] ; il est implémenté en Prolog et contient un module

pour la microplanification et une grammaire pour la génération superficielle. Le module pour la microplanification est spécialisé dans la tâche d'agrégation ; il ne réalise pas d'autres tâches spécifiques telles que la génération des expressions référentielles (seule une forme de pronominalisation des descriptions existe).

L'entrée au système ASTROGEN consiste dans une ou plusieurs « f-structures », c'est-à-dire des simples formules logiques contenant un prédicat et ses arguments, auxquels on ajoute le temps du verbe.

La sortie de ASTROGEN dépend des réglages faits au préalable ; le système « connaît » plusieurs règles d'agrégation, une règle de pronominalisation utilisable au choix. Le système ASTROGEN, implémenté en Prolog, contient une grammaire et un lexique pour un fragment réduit de l'anglais, utilisables pour essayer le système. Le système permet une forme très limitée de sensibilité au contexte, au sens où, en fonction des différentes caractéristiques de l'entrée (la présence des relations symétriques, des prédicats ou arguments partagés), des règles différentes d'agrégation sont appliquées pendant la génération.

En conclusion, ASTROGEN pourrait être approprié pour la génération de courts énoncés dans des dialogues oraux, étant quand même moins approprié pour la génération des énoncés plus longs, plus complexes.

Le système GEMINI

Le système GEMINI a été développé en tant qu'outil générique pour l'analyse (interprétation sémantique) du langage naturel. De plus, GEMINI contient un générateur basé sur le formalisme HPSG et qui utilise une grammaire bidirectionnelle employée à la fois pour l'analyse et pour la génération [170]. GEMINI a déjà été employé pour analyse et génération dans plusieurs systèmes de dialogue, comme par exemple CommandTalk et WTAS [170]. Ces derniers constituent des interfaces en langage parlé à un simulateur de champ de bataille et respectivement à un hélicoptère.

L'entrée dans le système consiste dans des formes logiques augmentées avec des informations pertinentes pour la génération, comme par exemple des détails concernant l'accord en nombre, le choix de l'article (indéfini / défini) et la génération des références.

Le système GEMINI en soi est réutilisable, mais les grammaires dont il se sert sont très dépendantes du domaine. Ce système a été particulièrement conçu pour l'emploi dans des systèmes de dialogue oral, c'est pour cette raison qu'il peut enrichir le texte produit avec des marqueurs prosodiques.

2.3.3 Générateurs textuels à base de grammaires

Deux des générateurs textuels les plus connus, KPML et FUF, sont présentés dans cette catégorie. Le troisième système, RealPro, est moins connu, étant donné qu'il est plus récent que les deux autres.

Le système KPML

Le générateur textuel KPML (« Komet-Penman Multilingual ») contient un moteur de génération, un ensemble assez important de ressources grammaticales et un environnement de développement assez riche [170]. Son « moteur » est basé sur le formalisme SFG (« Systemic Functional Grammar »), qui distingue les catégories linguistiques en termes de leur fonction plutôt qu'en termes de leur forme. Dans le cadre de la SFG l'idée fondamentale est de choisir entre des alternatives disponibles, ce qui la fait assez appropriée pour la tâche de génération. La réalisation linguistique dans le cadre de la SFG suppose le traversement d'un réseau systémique correspondant à une taxinomie d'éléments linguistiques en termes de leur fonction. Chaque divergence dans le réseau représente un choix entre des alternatives grammaticales, *via* l'ajout des contraintes supplémentaires sur la forme finale de l'énoncé généré.

En principe, le système KPML permet des choix fortement sensibles au contexte, en prenant en compte des aspects concernant les sens au niveau propositionnel, mais aussi au niveau interpersonnel (la relation entre le locuteur – la machine et l'allocutaire – l'utilisateur), ce qui pourrait déterminer le caractère plus ou moins formel des énoncés générés.

De plus, le système prend en compte des aspects discursifs (références anaphoriques, ellipses) afin de choisir parmi des constructions syntaxiques différentes qui ont le même sens. Cependant, toutes les informations concernant les choix en génération doivent être précisées dans la structure d'entrée ; sinon, le système opère les choix par défaut, non-sensibles au contexte.

Le générateur KPML, écrit en Common Lisp, est assez général ; il peut être utilisé pour la génération en différentes langues, telles que l'anglais, le français, le néerlandais, etc.

Le système FUF

Le formalisme FUF (« Functional Unification Formalism ») est un interpréteur pour un langage spécialement conçu pour le développement des applications visant la génération textuelle. En général le FUF est employé en tandem avec une grammaire exhaustive pour la langue anglaise, la SURGE [170]. Même si le FUF avait visé au début la couverture de toutes les tâches de génération, il est employé à présent pour la réalisation linguistique, travaillant de manière conjointe avec la grammaire SURGE.

La réalisation linguistique *via* l'utilisation d'une grammaire se fait, en FUF, en unifiant une structure d'entrée avec les structures trouvées *a priori* dans la grammaire. Le résultat de cette unification est représenté par une structure syntaxique qui est ensuite linéarisée afin de produire la chaîne orthographique qui correspond à l'énoncé textuel à générer. Le formalisme FUF emprunte des idées à la SFG, mais aussi à la HPSG.

Pareillement à KPML, en FUF la sensibilité au contexte dépend à la fois de la grammaire et du module de microplanification utilisé. Par ailleurs, en dehors de la sensibilité de la grammaire par rapport au contexte, toutes les caractéristiques sémantiques auxquelles le FUF est sensible doivent être précisées en entrée, offertes par le composant de microplanification.

Le FUF, implémenté en Common Lisp est particulièrement lent par rapport au KPML et même au RealPro, que nous décrivons par la suite. Cela le rend peut-être moins approprié pour les applications de dialogue.

Le système RealPro

Le générateur textuel RealPro est basé sur les grammaires de dépendances de Igor Mel'čuk, qui aborde le processus de génération sous forme d'une association itérative entre sept niveaux de représentation linguistique. Seulement quatre de ces niveaux sont implémentés en RealPro [170]. Des trois systèmes discutés dans cette section, RealPro est le plus simple, mais il requiert en même temps la spécification la plus détaillée en entrée.

Ainsi, l'entrée de RealPro consiste dans une structure sémantique dite « profonde », qui spécifie les lemmes des mots, leurs rôles syntaxiques et leurs caractéristiques ; cette structure ne contient quand même pas d'informations concernant l'ordre des mots dans l'énoncé.

L'entrée dans RealPro est assez détaillée : la plupart des choix d'ordre syntaxique et lexical avaient déjà été faits avant la génération proprement dite ; donc, ce que le système réalise est l'ordonnement des mots et l'application des contraintes morphologiques.

RealPro contient une grammaire et un lexique pour la langue anglaise ; ses relatives simplicité et modularité pourraient rendre le système approprié pour l'utilisation en dialogue ; de plus, par rapport aux autres générateurs textuels, celui-ci est assez rapide.

2.3.4 Générateurs textuels à base de structures figées

On va présenter ici, toujours de manière très succincte, deux systèmes effectuant la génération superficielle *via* une gestion plus ou moins élaborée des structures figées (en anglais, « templates ») : il s'agit de TG/2 et de YAG. Ces systèmes sont plus récents que les générateurs basés sur une grammaire, car c'est seulement pendant les dernières années que la communauté scientifique a commencé à apprécier la valeur pratique de cette approche, à la fois intéressante du point de vue théorique.

Le système TG/2

Le TG/2 est un générateur textuel basé sur des structures figées qui permet de varier la granularité de modélisation, conformément aux demandes de l'application courante.

Le TG/2 peut recevoir plusieurs types d'entrées, en les convertissant toutes dans une structure, exprimée à son tour dans un langage dédié, GIL (« Generator Internal Layer »).

Ce qui est intéressant à ce système c'est qu'il permet de combiner une approche grammaticale avec les structures figées, ces dernières utilisées par exemple pour des morceaux de texte spécifiques à une certaine application, tandis que la grammaire peut gérer les expressions plus sensibles au contexte (en termes d'utilisateur, etc.) et indépendantes du domaine (comme par exemple, les expressions temporelles).

La sensibilité au contexte réside dans l'utilisation des règles de sélection pour chaque structure figée. Donc, ce système pourrait être utilisé dans des applications de dialogue à domaine restreint et possibilités expressives relativement « larges ».

Le système YAG

Ce système est un générateur superficiel d'utilisation générale, basé sur des structures figées qui embarquent plusieurs types d'expressions de contrôle ou bien d'autres structures figées, de

manière récursive.

A l'instar des autres générateurs textuels, YAG ne peut pas gérer soi-même le contexte, mais il dépend des spécifications dans la structure d'entrée.

Il y a deux types de bases de connaissances dans YAG : les connaissances dépendantes du domaine et celles indépendantes du domaine. Les connaissances indépendantes du domaine contiennent une bibliothèque avec des structures figées qui peuvent être utilisées en tant qu'une sorte de « grammaire ». Les connaissances indépendantes du domaine contiennent aussi un lexique et des règles pour obtenir des formes fléchies pour les verbes. Les connaissances dépendantes du domaine sont représentées par des tableaux pour associer les représentations des connaissances spécifiques au domaine, aux structures figées qui les représentent du point de vue linguistique.

Une autre caractéristique importante qui rend YAG approprié pour le dialogue est le fait qu'il fonctionne en temps réel.

2.3.5 Générateurs « complets » à base de grammaire

Dans la catégorie des systèmes de génération basés sur une grammaire et qui gèrent à la fois la planification documentaire et sententielle (en fait, au moins et surtout la microplanification) et la réalisation linguistique, nous pouvons mentionner le SPUD (réalisé par Stone [166]) et le InDiGen (réalisé par Striegnitz [168]). Ces deux systèmes réalisent à la fois la microplanification et la génération purement textuelle, *via* des grammaires LTAG (« Lexicalized Tree Adjoining Grammar »). Ces systèmes prennent en entrée la spécification d'un « but communicationnel » produit par le contrôleur de dialogue ; cette spécification a la forme d'une représentation sémantique et, éventuellement pragmatique (actes de langage), sans détails syntaxiques. Donc, bien que ces systèmes ne gèrent pas la détermination du contenu communicationnel, ils sont beaucoup plus « vastes » que les simples générateurs textuels, c'est pour ça que nous les appelons aussi des systèmes « complets ».

Dans les systèmes SPUD (« Sentence Planning Using Descriptions ») et InDiGen, la génération de langage se déroule de manière bien motivée du point de vue théorique, étant basée sur des principes mis au point, aux niveaux syntaxique et sémantique, dans le formalisme LTAG. De plus, les arbres LTAG *augmentés* employés en SPUD et InDiGen sont très appropriés pour la génération sensible au contexte [170]³, car ils spécifient non seulement les formes de surface, mais aussi les contraintes sémantiques et pragmatiques sur l'usage de ces arbres ; il s'agit donc d'un modèle assez riche du contexte. A ce dernier égard, ces arbres augmentés ressemblent aux structures figées employées en TG/2, LGM et EXEMPLARS.

Ainsi, ces deux systèmes SPUD et InDiGen sont conçus pour résoudre des problèmes complémentaires en génération : tandis que le SPUD essaye de spécifier de manière aussi rigoureuse que possible le côté logique (inférentiel) touchant surtout aux aspects pragmatiques de la microplanification, le système InDiGen tente de résoudre de manière aussi située du point de vue théorique que possible le côté « linguistique » lié à la génération des références et de certains déictiques en microplanification. Nous voyons ainsi que les deux systèmes apportent les contri-

³Voir aussi les travaux de Stone [167] pour une discussion synthétique sur l'application de SPUD à des tâches de microplanification.

butions les plus spectaculaires à la génération vue du côté de la microplanification, vers les aspects pragmatiques (SPUD) et vers les aspects linguistiques (InDiGen).

2.3.6 Générateurs « complets » à base de structures figées

Peu de systèmes complets de génération ont été rendus publics, puisqu'un système complet de génération comporte en général une forte dépendance du domaine d'utilisation. Cela est vrai surtout pour les systèmes basés sur des expressions figées, qui doivent être redéfinies pour chaque nouvelle application. En conséquence, un tel système est utile à diffuser seulement s'il est accompagné par une documentation détaillée. Cependant, la plupart des systèmes de génération n'arrivent pas à une telle généralité qui puisse les rendre intéressants dans des contextes différents.

Il y a quand même deux exceptions à cette « règle » : le système LGM (« Language Generation Module »), développé à l'Université de Eindhoven par Theune [169] et le système EXEMPLARS, développé à CoGenTex [170]. Ces deux systèmes utilisent les structures figées pour la génération.

Les deux systèmes LGM et EXEMPLARS dénomment ces structures « structures syntaxiques figées » (« syntactic templates ») et, respectivement, « exemplaires » (« exemplars »). Les structures figées ont l'inconvénient d'être fort dépendantes au domaine, offrant en revanche l'avantage d'une génération efficace au niveau du temps de calcul. Ces deux systèmes ont encore en commun le fait d'être des systèmes complets de génération, gérant ainsi à la fois les tâches de planification discursive et de microplanification, et la génération textuelle proprement dite.

Ces points communs mis à part, les deux systèmes se représentent différemment les structures figées. En LGM il y a une association directe, biunivoque entre l'entrée et l'ensemble des structures figées, *via* un tableau de correspondances, ce qui pourrait être efficace, mais ne permet pas de variations. Dans les générateurs textuels « purs » les expressions figées sont choisies à partir des conditions (règles) d'applicabilité associées, lorsqu'il y a plus d'une structure figée qui peut être appropriée pour une certaine entrée. De cet ensemble de structures figées applicables, EXEMPLARS choisit toujours la structure figée la plus spécialisée et LGM fait un choix aléatoire.

En ce qui concerne l'ajout des verbes aux structures figées, les systèmes TG/2 et YAG (présentés ci-dessus) utilisent des règles générales pour obtenir des formes fléchies pour les verbes, tandis que LGM n'a pas de telles règles ; ce dernier détermine de manière *ad-hoc* les formes fléchies pour les verbes, à l'intérieur des structures figées. Dans le système EXEMPLARS, les exemplaires ne traitent non plus les flexions des verbes ; ici, le temps des verbes est codé de manière explicite dans les exemplaires (les structures figées de ce système).

Les structures figées de LGM contiennent des arbres syntaxiques complètement spécifiés pour un certain énoncé à être généré (d'ici le nom de « structures figées syntaxiques »). Ces informations sont utilisées pour le calcul de la prosodie. Les exemplaires d'EXEMPLARS ne contiennent pas d'informations syntaxiques. De plus, les systèmes complets de génération basés sur des structures figées, tels que LGM et EXEMPLARS, ont l'avantage que l'information contextuelle nécessaire pour le choix des expressions figées est produite par ces systèmes-mêmes. Conséquemment, l'utilisation des systèmes LGM et EXEMPLARS en dialogue serait possible, en combinaison même avec un contrôleur de dialogue moins complexe.

2.3.7 Conclusions

Nous avons présenté plusieurs systèmes de génération du langage naturel, plus ou moins sophistiqués et plus ou moins complets. De ces systèmes, certains seraient, à notre avis, plus appropriés pour l'emploi en génération dialogique, que les autres. Ainsi, les systèmes basés sur des grammaires réversibles (ALE, ASTROGEN, GEMINI) sont en général assez « figés » au sens où ils n'offrent pas trop de flexibilité et ne prennent pas en compte de manière satisfaisante le contexte d'utilisation.

Ensuite, il y a les générateurs purement textuels, réalisant seulement les tâches liées à la génération linguistique ; ceux-ci se divisent à leur tour en générateurs textuels à base de structures figées (TG/2 et YAG) où à base de grammaire (KPML, FUF, RealPro) et ne gèrent pas eux-mêmes les aspects contextuels ou la microplanification ; il faut que l'entrée dans ces systèmes soit suffisamment détaillée pour qu'ils puissent offrir des réponses naturelles en dialogue.

Finalement, nous avons vu un ensemble de systèmes de génération complets, gérant en principe tous les niveaux standard en génération. Ces systèmes se divisent à leur tour en systèmes basés sur des structures figées (LGM et EXEMPLARS) et systèmes basés sur une grammaire (SPUD et InDiGen). Ces systèmes complets offrent l'avantage de pouvoir être utilisés avec des contrôleurs de dialogue assez peu complexes et / ou de pouvoir gérer de manière rigoureuse le contexte, offrant en même temps une flexibilité importante. Néanmoins, ces systèmes ont quelques inconvénients pour l'utilisation en dialogue, tels que la nécessité d'en supprimer certains composants (la planification du contenu, par exemple) ou de modifier d'autres (par exemple, la microplanification doit être adaptée pour prendre en compte le caractère discontinu et séquentiel des énoncés en dialogue – une phrase peut couvrir plusieurs tours de parole, etc.).

Bref, compte tenu des exigences auxquelles un générateur devrait satisfaire pour être utilisable en dialogue, nous croyons qu'un système complet de génération pourrait être meilleur, car il assure en même temps une sensibilité importante au contexte d'utilisation et une autonomie par rapport aux autres modules d'un système de dialogue, ce qui augmente la ré-utilisabilité du générateur.

Quant au choix entre l'utilisation des structures figées ou d'une grammaire, cela dépend des nécessités concrètes : si les contraintes de temps ne sont pas critiques et on souhaite une dépendance du domaine aussi basse que possible, on peut choisir l'utilisation d'une grammaire ; en revanche, si les contraintes de temps sont importantes et si on souhaite utiliser le générateur dans le contexte d'une application donnée, alors les expressions figées pourraient être choisies. Il n'en est pas moins qu'on a la possibilité de combiner les approches, comme dans le générateur de l'agent conversationnel TRIPS, que nous décrivons dans le paragraphe suivant.

2.4 Générateurs conçus pour le dialogue homme-machine

2.4.1 Le générateur de Stent

Vue d'ensemble

Les recherches de Stent concernent la génération de contributions dialogiques aussi « naturelles » que possible pour des systèmes de dialogue oral homme-machine finalisés par la tâche.

Plus précisément, il s'agit du composant de génération de la parole dans le système TRIPS, développé à l'Université de Rochester. Ce système de dialogue utilise des modèles riches pour le contexte de la tâche et pour les intentions de l'utilisateur. Ainsi, TRIPS peut gérer des dialogues complexes dans un ensemble très varié de domaines, tels que la planification, l'assistance *via* des conseils, etc.

Derrière la génération de profondeur, c'est-à-dire, la planification pragmatique du contenu des contributions dialogiques à générer par le système, il y a la théorie des actes de dialogue de Traum et Hinkelman [177], [165]. Ainsi, on suppose qu'une hiérarchie de comportements dialogiques, ou *actes conversationnels* existe, tels que : *la prise des tours de parole, l'établissement de l'arrière-plan commun* pour le dialogue (en anglais, « *grounding* »), *les actes de langage* (dans le sens de Searle) et *les actes d'argumentation*. Nous observons donc que selon la théorie de Traum et Hinkelman il existe une hiérarchie d'actes de conversation, dont les actes de langage ne représentent qu'un niveau.

De manière plus détaillée, les prises des tours de parole précisent quel locuteur peut parler à tout point dans la conversation. L'établissement de l'arrière-plan commun s'assure que tous les locuteurs sont « mis à jour », au sens où chaque locuteur connaît ce qu'il est connu et cru par les autres partenaires de dialogue. Les actes de langage ont leur sens traditionnel, d'actions réalisées *via* le langage. Enfin, les actes d'argumentation constituent des ensembles d'actes de langage trouvés dans une certaine relation, soit en termes fonctionnels, soit en termes de paires adjacentes.

Ce qui est de fondamental ici est que, selon la Théorie des actes conversationnels de Traum et Hinkelman, la réalisation des actes aux niveaux plus bas (les prises des tours de parole sont au plus bas niveau) est une condition nécessaire pour la réalisation des actes aux niveaux plus hauts. Par conséquent :

- étant donné que la réalisation des actes aux niveaux plus hauts (par exemple, les actes de langage) est conditionnée par la réalisation des actes aux niveaux plus bas (par exemple, les prises des tours de parole), les actes aux niveaux bas ne sont pas forcément réalisés en langage naturel, c'est-à-dire, un acte à un niveau bas peut être réalisé de manière intégrée dans la réalisation d'un acte de plus haut niveau ;
- si un acte de plus bas niveau est réalisé en tant que tel, il doit précéder la réalisation d'un acte de plus haut niveau qui aurait pu l'intégrer (au sens précisé ci-dessus), donc, logiquement, l'acte de plus bas niveau ne peut pas être réalisé après l'acte de plus haut niveau ; on désigne par *contrainte d'ordonnement* ce phénomène.

Stent est, selon sa propre assertion [165], le premier chercheur à avoir utilisé la théorie des actes conversationnels pour la génération dans le contexte du dialogue oral. Ainsi, les questions que Stent s'est posées lors de la génération basée sur la théorie des actes conversationnels sont les suivantes :

- étant donné que la théorie des actes conversationnels a été employée avec succès pour l'analyse du discours, dans quelle mesure peut-elle offrir un support cohérent pour la formalisation des comportements humains dans des dialogues réels ?
- pourrait-on utiliser les concepts de base de la théorie des actes conversationnels pour construire un système informatique de génération de la parole pour le dialogue oral, ayant une sortie comparable aux performances des sujets humains ?

Du point de vue de nos propres recherches, ce qui nous intéresse est plutôt la deuxième question, car des recherches antérieures avaient déjà démontré que la réponse à la première question est affirmative, par des études sur des corpus de dialogues [165].

Les idées fondamentales de la théorie des actes conversationnels qui sont mises en valeur lors des démarches de Stent sont :

- le dialogue oral comprend plusieurs types de comportements (ou types d'actes), tels que la prise des tours de parole, l'établissement de l'arrière-plan commun, les actes de langage et l'argumentation ;
- ces comportements sont organisés de manière hiérarchique, les actes de bas niveau contribuant à la réalisation des actes de haut niveau.

Une autre supposition importante de la Théorie des actes conversationnels de Traum et Hinkelman est que ces comportements dialogiques sont constitués d'actions discrètes réductibles à des actes et qui peuvent être planifiées utilisant des techniques de planification traditionnelles dans l'intelligence artificielle.

Le système de dialogue utilisé par Stent lors de ses expérimentations et réalisations informatiques est, comme nous l'avons déjà annoncé, TRIPS. Ce système consiste dans un ensemble de composants hétérogènes qui partagent des informations *via* des messages en KQML. L'architecture logique du système TRIPS est décrite dans [164], [165] et, ne concernant pas à vrai dire l'état de l'art sur la génération en soi, ne sera pas détaillée ici.

La génération textuelle dans le système TRIPS est de la responsabilité d'un module dédié, appelé « Gestion de la Génération » qui met en correspondance les intentions communicationnelles fournies par le contrôleur de dialogue (appelé ici « Agent Comportemental ») avec des actes conversationnels⁴ et organise ces derniers en contributions langagières cohérentes et naturelles. Ce processus est semblable à la structuration du contenu en ce qu'il s'agit de l'organisation des éléments fondamentaux dans une structure discursive cohérente. Ce processus est appelé par Stent *Planification des contributions*.

De même, le générateur (« Gestion de la génération ») est responsable de la transformation du contenu dans des formules logiques, dont chacune peut être réalisée, dans la plupart des situations, *via* un seul énoncé. Cependant, c'est le générateur textuel (appelé de même par Stent) qui doit ordonner les informations dans l'énoncé, choisir les objets lexicaux et résoudre les références⁵.

Planification du contenu

L'entrée dans le processus de planification des contributions dialogiques est représentée par un ensemble d'intentions communicationnelles, éventuellement avec un certain contenu associé.

⁴Voici une différence importante par rapport à l'architecture de Caelen [35], adoptée par nous-même dans cette thèse, où le contrôleur de dialogue établit lui-même l'acte de *langage* à générer, tandis que le générateur doit régler la force illocutoire pour celui-ci, le rendre cohérent par rapport à l'historique du dialogue, le mettre sous forme textuelle et ensuite orale.

⁵Voici, à nouveau, une autre différence par rapport à l'architecture de Caelen : la génération des formules logiques est la responsabilité conjointe du contrôleur de dialogue et du générateur de profondeur (planificateur pragmatique, voir plus loin), tandis que le générateur textuel (linguistique) peut opérer certains traitements sur les formules logiques reçues en entrée.

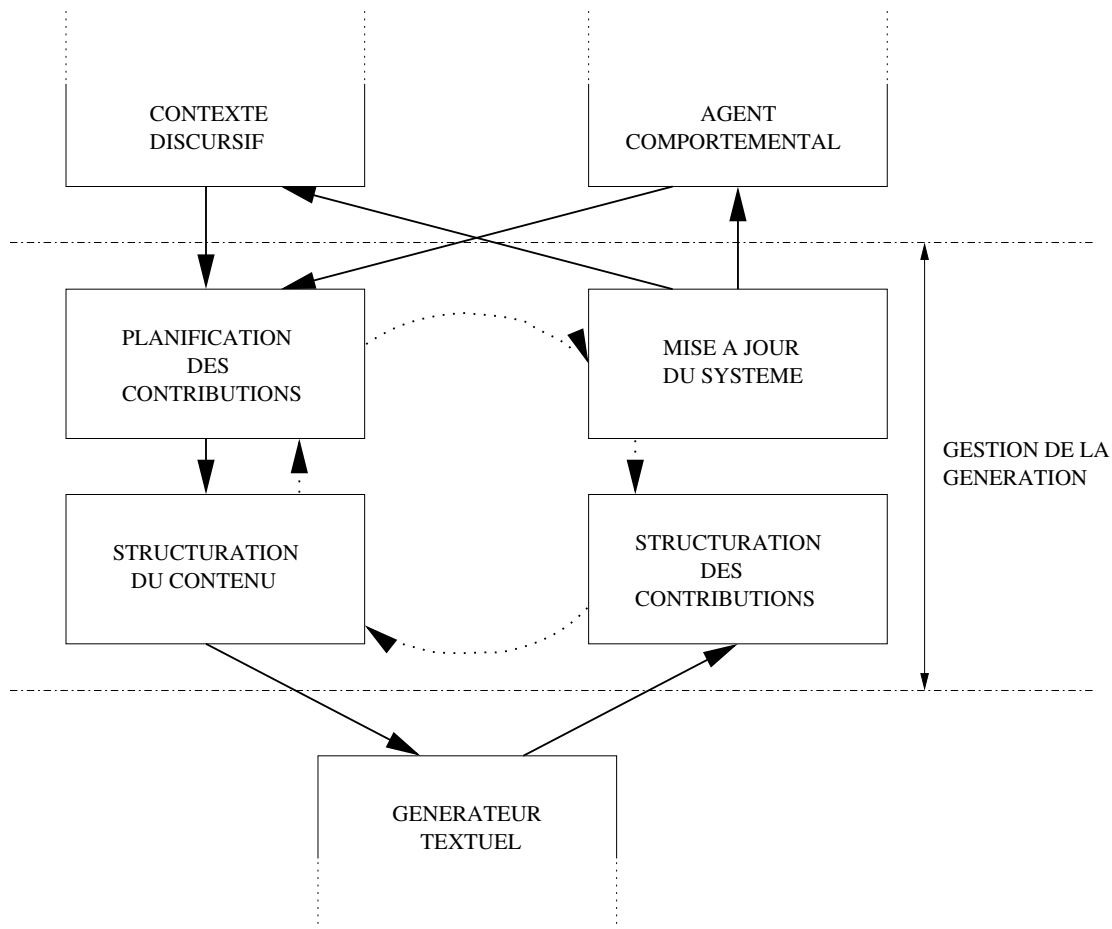


FIG. 2.1 – Architecture générale du générateur profond de Stent

Ces intentions communicationnelles ne sont cependant générées qu’au fur et à mesure que le dialogue avance dans le temps et lors du traitement des incompréhensions éventuelles, surtout au niveau des intentions à l’égard du discours en tant que tel.

Ce qui semble important pour Stent est que le générateur profond (Gestion de la génération) n’attend pas un signal pour commencer à planifier, car, du point de vue pratique, le système a très peu de temps pour commencer à répondre, après que l’usager ait fini son énoncé. En conséquence, le contrôleur de la génération doit planifier constamment, en s’appuyant sur ce qu’il connaît à chaque instant, à la manière dont un usager humain planifie ses contributions dialogiques [102].

L’architecture du contrôleur de la génération est présentée dans la figure 2.1, où chaque nouvelle intention communicationnelle ou chaque mise à jour de l’état du discours amorce la planification des contributions dialogiques, qui peut mener à son tour à la planification d’un ou plusieurs actes conversationnels. Une deuxième étape, appelée ici « Structuration de la contribution », réalise un filtrage et un ordonnancement de la sortie du module de génération textuelle

et décide quels actes seront produits en fin de compte. Chaque étape des traitements réalisés par le contrôleur de la génération met en valeur des aspects différents de la théorie des actes conversationnels : la planification des contributions met en correspondance les intentions communicationnelles avec des niveaux de comportement dialogique et actes conversationnels individuels, tandis que la structuration des contributions s'appuie sur les relations d'intégration (d'un acte de niveau bas à un acte de niveau plus haut) et d'ordonnement entre les actes conversationnels à des niveaux différents.

L'unité de base dans l'état du contrôleur de la génération est la contribution courante qui a, en général, la longueur d'un seul tour de parole mais qui peut s'étendre sur plusieurs tours de parole si les énoncés provenant de l'utilisateur ne changent pas l'état du système, au niveau de la *résolution des problèmes* – pour plus de détails, voir [165]. Le contrôleur de la génération gère, pour chaque contribution dialogique, les éléments suivants :

- tous les énoncés provenant de l'utilisateur auxquels le système répond dans la contribution en question ;
- toutes les intentions communicationnelles déterminées par les énoncés de l'utilisateur et qui pourraient être satisfaites dans la contribution en question ;
- toutes les intentions communicationnelles reliées à la résolution du problème courant et qui pourraient être satisfaites dans la contribution en question ;
- tous les actes conversationnels en sortie, qui pourraient être ou avaient été produits dans la contribution en question.

De plus, il y a de fortes liaisons entre tous ces éléments et les éléments correspondants des autres énoncés et / ou contributions associé(e)s au problème courant.

La planification du contenu décide, pour une intention communicationnelle donnée, quels actes conversationnels utiliser afin d'augmenter l'intention vers sa mise en forme linguistique. Pour cela, on utilise un ensemble de règles qui modélisent les divers types d'actes conversationnels, dont la plupart sont des actes d'argumentation.

A chaque fois que l'état des contributions est mis à jour, la planification de la contribution trouve toutes les règles appropriées et les attribue un poids selon le degré dont les conditions de la règle « recouvrent » l'état des contributions.

Il n'y a pas de retours en arrière lors de la planification des contributions ; si le générateur textuel n'arrive pas à calculer les formes de surface, alors on ne choisit pas une nouvelle règle, du fait que l'état des contributions peut changer fortement dans le temps, *via* l'ajout de nouveaux buts, de nouveaux énoncés planifiés, etc.

Quand l'acte conversationnel a été choisi en lui attribuant un contenu, on doit transformer ce contenu dans une formule logique. Pour cela, le sous-module de structuration de contenu sur la figure 2.1 utilise la correspondance entre le domaine de la tâche et les informations pour la planification (actions, états, objets) et les représentations sémantiques orientées rôle que le TRIPS emploie.

Chaque acte conversationnel, avec le contenu associé, est envoyé au générateur textuel ; si ce dernier répond avec une forme de surface, alors l'état des contributions est mis à jour. Les formes de surface sont reliées aux actes conversationnels qu'elles réalisent. Ensuite, le processus de structuration des contributions est invoqué.

Le processus de structuration des contributions utilise un ensemble de règles précisant la

manière dont les différents niveaux des actes conversationnels interagissent et la manière dont les actes conversationnels à un même niveau peuvent être organisés. Le module de structuration des contributions a une certaine liberté concernant le choix des formes de surface à produire, étant donnée la relation d'intégration existante entre les différents niveaux des actes conversationnels.

Chaque règle dans ce module a un certain nombre de contraintes qui peuvent être classifiées ainsi : contraintes temporelles, contraintes sur l'acte courant situé en focus, contraintes sur d'autres actes dans le tour de parole courant. A leur tour, les contraintes temporelles sont de deux types : contraintes sur le temps depuis quand l'utilisateur a cessé de parler et contraintes sur le temps écoulé depuis la dernière contribution dialogique du système. Ces contraintes temporelles sont nécessaires pour modéliser les aspects dépendants de temps dans l'alternance des prises des tours de parole. Les contraintes applicables à un acte de conversation incluent des contraintes sur le niveau de l'acte, sur le type de l'acte, et sur la production ou non-production de l'acte.

Mise à jour de l'état discursif

Lorsque la production d'un énoncé, par le système, est complète, le contrôleur de la génération reçoit une mise à jour de l'état de la composante de génération textuelle. A l'instar de la manière dont TRIPS traite les énoncés provenant des usagers, on envoie aux contexte discursif la structure syntaxique et la forme logique.

Génération textuelle

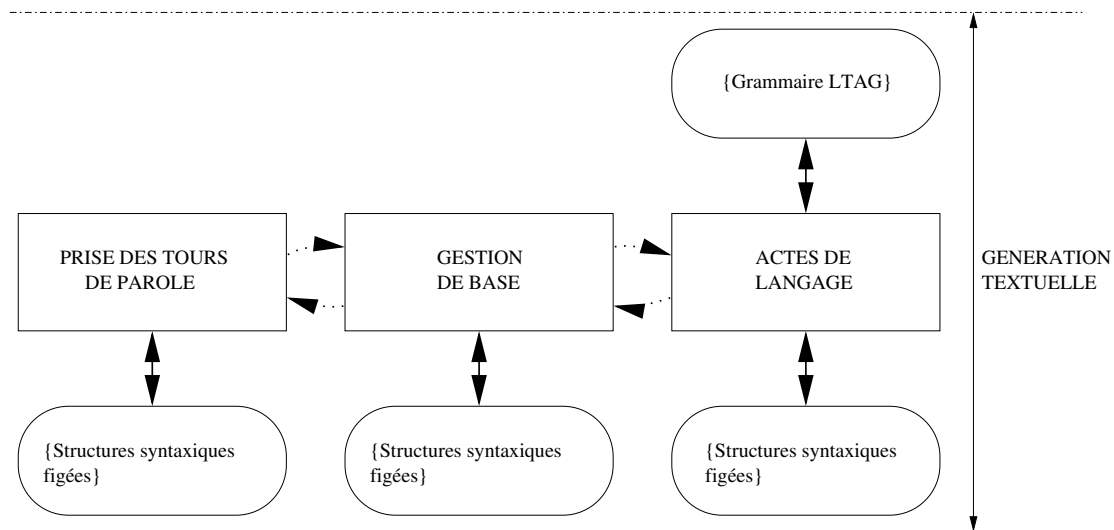


FIG. 2.2 – Architecture générale du générateur textuel

Le générateur conçu par Stent remplace l'ancien générateur de TRIPS, basé sur des constructions syntaxiques figées. L'approche choisie par la chercheuse américaine essaye de combiner la portabilité et l'efficacité d'exécution, *via* l'utilisation parallèle des constructions syntaxiques figées (pour les prises des tours de parole et pour l'établissement de l'arrière-plan commun et

en partie pour les actes de langage) pour l'efficacité et d'une grammaire TAG (« Tree Adjoining Grammar ») lexicalisée, approche standard dans la génération du langage naturel (pour la plupart des actes de langage) pour la généralité et la flexibilité. Cette approche est montrée de manière synthétique dans la figure 2.2.

Le générateur textuel fait, lui aussi, de la planification de la phrase, en termes d'ordonnement de l'information sur la forme logique ou d'ajout de certaines valeurs par défaut pour quelques concepts censés être générés. Il s'appuie, lui aussi, sur le contexte discursif pour la résolution des références.

Après ce traitement initial de la forme logique en entrée, le générateur textuel communique aux modules adéquats les actes de conversation, la forme sémantique (logique) et les marqueurs syntaxiques.

La génération superficielle des tours de parole et la gestion de l'arrière-plan commun sont réalisées *via* des expressions figées, car la production de ces types d'actes ne nécessite pas la construction et la représentation de structures complexes qui reflètent des formules logiques complexes. De plus, ces actes doivent être produits *en temps réel*, d'où la nécessité que la génération de ces actes soit aussi efficace que possible.

En revanche, pour la génération de la plupart des actes de langage, Stent utilise une grammaire LTAG. L'utilisation d'une grammaire permet de gérer des aspects sémantiques et syntaxiques dont la complexité ne permet pas l'utilisation satisfaisante des expressions figées.

L'algorithme de génération pour tous les actes conversationnels est, essentiellement, une recherche en profondeur (« depth-first ») de haut en bas avec retour arrière, sur l'arbre sémantique (TAG).

Conclusions

Les recherches de Stent ont essayé de trouver des manières dont on peut améliorer le caractère « naturel » des contributions dialogiques générées par la machine, dans le contexte des dialogues complexes, finalisés par la tâche.

La théorie des actes conversationnels de Traum et Hinkelman a été choisie pour ce but et a été implémentée dans le contexte des composantes de génération dans le système TRIPS de dialogue homme-machine. Les conclusions de ces recherches ont été que la théorie des actes conversationnels marche bien en tant que modèle de la génération des actes pour la gestion de l'arrière-plan en dialogue et pour les actes de langage. Une idée-clé de cette théorie, selon laquelle certains types d'actes peuvent être intégrés aux autres s'est avérée un élément important dans l'amélioration du caractère naturel des énoncés générés. De plus, le fait que la théorie des actes conversationnels contribue à clarifier la relation entre les intentions communicationnelles et les comportements dialogiques a aidé Stent à créer des composantes de génération très flexibles, dans le cadre du TRIPS.

Cependant, Stent a constaté que lorsqu'il s'agit des deux autres niveaux des actes de conversation, il y a certaines difficultés ; en particulier, la prise des tours de parole n'est pas décrite de manière satisfaisante dans le cadre de cette théorie. C'est pourquoi un aperçu rhétorique de ces phénomènes serait, à notre avis, nécessaire, comme le reste des chapitres dans cette thèse le montrera.

2.4.2 Générateur de Jokinen

Vue d'ensemble

Le système de dialogue développé par Jokinen [86] essaye de répondre à certains enjeux du projet européen ESPRIT PLUS, dont le but était de construire une interface robuste et extensible à la version électronique des Pages Jaunes en anglais (« Yellow Pages »). Ce projet a étudié le rôle du contexte et du raisonnement sur celui-ci, dans l'interaction homme-machine. Le principe régissant a été de reposer sur les connaissances et le raisonnement pragmatiques afin d'atteindre le plus de robustesse possible.

Le travail de Jokinen est une extension et une élaboration de ce type d'approche basé sur la pragmatique, dans la gestion du dialogue. Il concerne les *desiderata* d'un système de dialogue coopératif et flexible, et présente un prototype d'un système qui traite de la robustesse, du point de vue de la compétence communicationnelle. La nouvelle approche pour la modélisation du traitement de l'information dans ce système de dialogue est dénommée *Gestion constructive du dialogue*. Les recherches de Jokinen concernent notamment la planification et la génération des réponses du système, aussi que les principes communicationnels qui régissent ces processus.

La base théorique de ces travaux réside dans les concepts de coopération idéale, rationnelle et motivée, en tant que développés par Allwood [2]. Cependant, les idées de Allwood ont servi de source d'inspiration, plutôt que de base formelle entièrement développée. Le travail de Jokinen est original au sens où à partir des principes de Allwood il essaye de fonder les *desiderata* d'un contrôle du dialogue qui soit coopératif et robuste, sur une théorie pragmatique unifiée. Ce travail a été le premier à spécifier, formaliser et implémenter les idées de Allwood dans un système prototype de dialogue homme-machine.

Selon les principes de Allwood de gestion constructive de dialogue, la communication est vue comme une activité coopérative entre des agents rationnels. Les agents analysent et évaluent les contributions de leurs partenaires et fournissent en retour le résultat de cette évaluation. Cependant, ils n'utilisent pas le langage seulement pour atteindre leurs propres buts dans une activité à une seule direction, mais ils sont engagés dans une action conjointe : pendant qu'il élabore son propre but, le locuteur doit indiquer le fait que la réponse prend aussi en compte les attentes que le partenaire avait souhaité exprimer et évoquer par sa contribution. Cela est vrai parce que la communication crée des obligations sociales et normatives qui déterminent les agents à agir selon les principes de la « coopération idéale » (selon lesquels ils ont la volonté de continuer le dialogue jusqu'à ce que tous les deux aient atteint leurs buts et tous les deux soient satisfaits par le résultat). Certainement, l'interaction humaine implique aussi des conflits, mais selon Allwood, si le conflit devient si grave qu'il rend la coopération impossible, alors la communication échouera aussi.

Le fondement empirique des travaux de Jokinen repose sur la collecte de corpus réalisée dans le projet PLUS. Le but de cette recherche n'était pas de modéliser les dialogues humains en tant que tels, mais plutôt d'étudier quel type de phénomènes apparaissent dans les dialogues de recherche d'informations, entre un humain et une machine, et ensuite, à partir des consignes de la théorie de Allwood, de concevoir un système capable de répondre aux requêtes des usagers, d'une manière robuste et adéquate. Cela requiert une certaine idéalisation et généralisation sur les échantillons de dialogues obtenus empiriquement. Beaucoup d'aspects, tels que les pauses,

les chevauchements, les interruptions extérieures, les conversations à plusieurs locuteurs, l'influence des rôles sociaux des interlocuteurs, ne sont pas pris en compte dans ce travail de recherche.

En conclusion, les buts de la recherche de Jokinen concernent plusieurs aspects :

- étudier le rôle de la pragmatique dans l'amélioration de la robustesse en contrôle du dialogue ;
- utiliser les idées de Allwood sur la coopération idéale et activité motivée rationnellement, afin d'atteindre une gestion de dialogue à la fois robuste et efficace ;
- appliquer ces idées à la planification des réponses, notamment pour déterminer et contrôler la cohérence thématique de celles-ci et l'ellipse de certaines parties de ces réponses ;
- implémenter ces consignes dans un prototype informatique.

Il faut toutefois préciser que le travail de Jokinen mélange contrôle du dialogue et planification des réponses, donc les aspects relevant de la gestion du dialogue ne sont pas nettement séparés de ce qui tient à la génération proprement dite.

Regard critique : apport de la pragmatique en gestion du dialogue

Les contributions du travail de Jokinen s'articulent autour de plusieurs aspects ; ainsi, on identifie d'abord les caractéristiques de la robustesse, pour les dialogues de recherche d'informations, pour ensuite formaliser ces dernières en tant que contraintes sur la gestion du dialogue. Les idées de Allwood sur la communication vue comme une activité coopérative rationnelle sont appliquées à l'interaction homme-machine, afin d'aboutir à une base unifiée pour la gestion des dialogues coopératifs.

On peut remarquer que le travail de Jokinen ne fait pas une séparation nette entre la gestion proprement dite du dialogue (ce que nous appelons « contrôle de dialogue » – voir chapitre 1) et la production des répliques de la machine. Plus précisément, dans le système de Jokinen, la responsabilité sur toutes les consignes concernant les caractéristiques pragmatiques (ou contextuelles) des énoncés revient au contrôleur de dialogue. Le générateur de réponses n'est qu'un ensemble d'énoncés prédéfinis, dont un sous-ensemble est choisi (toujours par le contrôleur de dialogue), lors de la production d'une réplique. Ainsi, nous pouvons considérer que son approche est plutôt proche, dans l'esprit, des travaux de Appelt en génération intégrée (c'est à dire, Appelt ne souscrit pas à l'architecture standard de Reiter et Dale en génération de langage [147], préférant des frontières plus floues, voire absentes, entre les différentes étapes en génération – planification du discours, microplanification et réalisation de surface [8]).

Il n'en est pas moins vrai que le travail de Jokinen offre des pistes valables pour l'intégration de la pragmatique aux sources d'information utilisées pour guider la production des répliques langagières en dialogue homme-machine ; l'idée de mettre l'accent sur des normes et contraintes sociales sur les énoncés qui peuvent être produits, étant donnée une intention quelconque, est, au fond, la même que celle assumée par nos recherches. Il y a toutefois une (assez) grande différence entre les travaux de Jokinen et les nôtres : tandis que Jokinen essaye d'établir un ensemble de contraintes pragmatiques à la gestion du dialogue *per se*, nous travaillons dans un cadre plus sophistiqué, au sens où le contrôle du dialogue est guidé par une modélisation stratégique inspirée de la théorie de jeux [35], [34], et la modélisation du passage de l'intention communicationnelle (calculée par le contrôleur de dialogue) à la réplique linguistique est guidée

par un autre ensemble de principes, présentés tout au long de cette thèse. De plus, Jokinen ne considère que les dialogues entre une machine et *un* utilisateur humain, et il ne nous semble pas évident d'étendre son cadre au dialogue à plusieurs locuteurs.

Conclusions

On voit donc comment le travail de Jokinen sur les contraintes que les normes « sociales » imposent aux tours de parole en dialogue rejoint des recherches contemporaines (de Kibble [91], [92]) sur le rôle des engagements sociaux dans le renforcement de la cohérence des dialogues. Cependant, pour ce qui tient à la génération proprement dite des répliques, le seul aspect qui nous paraît pertinent dans ses travaux concerne le contrôle des ellipses (voir chapitre 6 pour une discussion plus approfondie des travaux de Jokinen sur cet aspect), où, en modélisant les connaissances des interlocuteurs (à la fois celles *a priori* existantes et celles acquises en dialogue), on parvient à contrôler le degré dans lequel les énoncés produits sont explicites.

Au niveau de l'évaluation de l'approche proposée, Jokinen, comme nous même, manquons d'une véritable évaluation quantitative de l'efficacité des méthodes mises en œuvre : le manque de ressources (corpus de dialogue homme-machine, éventuellement à plusieurs locuteurs) annotées de manière appropriée (par exemple, au niveau des contenus sémantiques des répliques, des structures rhétoriques engendrées et des actes de langage réalisés *via* ces énoncés).

2.5 Conclusions

Dans ce chapitre, après avoir défini de manière précise ce qu'on entend par « génération du langage naturel », nous avons montré les problèmes posés par le dialogue à ce processus. Ensuite, nous avons dressé un panorama des techniques existantes en génération, tout en discutant leur adéquation et leur pertinence dans les applications de dialogue homme-machine. Nous avons donc observé qu'aucune technique de génération ne peut être appliquée telle quelle au dialogue, surtout lorsqu'on prend en compte les contraintes d'interactivité et de généricité par rapport à la tâche. Par conséquent, deux approches se sont contournées jusqu'à présent pour aborder la génération des énoncés en dialogue homme-machine :

1. insérer un niveau de traitement « pragmatique » entre la gestion du dialogue et la génération linguistique des répliques ; une des recherches les plus représentatives à cet égard est celle de Stent, où la théorie des actes conversationnels de Traum et Hinkelman en constitue la clé de voûte ;
2. intégrer ces traitements pragmatiques au contrôleur du dialogue, pour que ce dernier puisse prendre pratiquement toutes les décisions concernant la génération de la réponse, qui ne reste qu'à être directement mise sous forme linguistique, notamment en choisissant un patron syntaxique approprié dans un ensemble d'expressions figées à trous pour une intention communicationnelle précisée. Une des recherches les plus représentatives à cet égard est celle de Jokinen, où les principes de Allwood de gestion constructive du dialogue contraignent le choix de la forme linguistique appropriée.

Nos propres recherches s'inscrivent plutôt sur la voie 1 ci-dessus, en ajoutant une dimension rhétorique à ce composant pragmatique. Cela est motivé par les limites inhérentes des approches

antérieures, concernant notamment le réglage fin des relations entre les énoncés, et par la nécessité de définir un cadre à la fois souple et générique par rapport à la tâche et même, dans une mesure importante, à la langue, pour piloter l'arrivée de l'intention communicationnelle à l'énonciation effective d'un tour de parole.

Chapitre 3

Structuration du discours en génération : filtrage local des relations rhétoriques

3.1 Introduction

Comme nous l'avons déjà mis en évidence dans les chapitres antérieurs, la génération des énoncés en dialogue est fortement dépendante du contexte. Cependant, relativement peu de systèmes de dialogue manipulent ce contexte pour piloter la génération de leur répliques. Une des raisons pour cet état des choses réside dans la complexité importante de modéliser, représenter et utiliser ce contexte. Si on se limite à des domaines très restreints (par exemple, réservation des vols [117]), on peut toujours atteindre un niveau satisfaisant pour les interactions homme-système, en ignorant ce contexte. Pourtant, pour des applications plus complexes, comme l'emprunt des livres dans une bibliothèque, où le système est censé aussi « connaître » quelques éléments sur le contenu des livres, il est nécessaire de prendre en compte l'historique du dialogue afin que les énoncés produits par la machine soient compréhensibles et plus naturels : par exemple, des constructions anaphoriques ou elliptiques devraient être produites à bon escient, des degrés de force appropriés devraient être attribués aux actes produits, et aussi des connecteurs discursifs pertinents du point de vue pragmatique devraient être choisis entre les diverses parties des tours de parole de la machine.

Tous ces traitements en génération peuvent être réalisés à partir d'une certaine représentation du contexte et de l'historique du dialogue. Plusieurs alternatives existent à nos jours pour cela : d'un côté, il y a les approches à base d'états informationnels (en anglais, « information state ») [179], ou des traitements plus orientés vers les aspects linguistiques, comme ceux basés sur des représentations du discours [111]. Les recherches décrites dans ce chapitre poursuivent cette deuxième voie, en utilisant une approximation, en logique du premier ordre, de la SDRT. Ce cadre permet de calculer l'ensemble des relations rhétoriques entre un énoncé courant et un énoncé antérieur, en prenant aussi en compte *les autres* énoncés dans l'historique du dialogue et les relations rhétoriques entre ces derniers.

Le processus de structuration rhétorique consiste, essentiellement, dans trois étapes : (i) filtrage pragmatique local des relations *potentiellement* valides (pour cela, nous utilisons les contraintes que les actes de langage imposent sur les relations rhétoriques possibles « entre » ces actes), (ii) filtrage sémantique local d'un ensemble de relations rhétoriques valides, à partir des relations fournies à l'étape précédente, et (iii) filtrage contextuel de la sortie de l'étape antérieure.

Dans ce chapitre nous allons décrire de manière détaillée les étapes (i) et (ii) ci-dessus. L'étape (iii) sera décrite *in extenso* dans le chapitre 4. Contrairement à l'enchaînement naturel des étapes, nous présentons tout d'abord le filtrage sémantique (étape (ii)), car cela nous permet de bien établir notre cadre théorique, pour ensuite décrire le filtrage pragmatique (étape (i)).

3.2 Une première contribution originale à la génération en dialogue : le problème du filtrage sémantique des relations rhétoriques

3.2.1 Position du problème

Au début, les systèmes de dialogue homme-machine se basaient sur des modules de génération conçus de manière *ad-hoc*, surtout à base de structures figées, pour produire les énoncés [116]. Cependant, dans la dernière décennie, avec l'émergence des résultats de recherche et des idées dans le domaine des systèmes multi-agents [52], [117], les systèmes de dialogue sont devenus plus sophistiqués, en visant de meilleures réponses aux requêtes des usagers, *via* des tours de parole plus naturels, en rapport avec le contexte du dialogue et avec les locuteurs impliqués [35], [116], [117]. Donc, la composante de génération du langage a commencé à évoluer pour fournir des productions langagières situées. Pour cela les aspects rhétoriques et actionnels du langage ont dû être pris en compte. Deux tendances ont commencé à se distinguer : (i) une approche basée sur la rhétorique, qui utilise des représentations formelles du discours, représentations conçues à l'origine pour l'interprétation du langage : ainsi, des théories comme la RST (« Rhetorical Structure Theory ») [107], la DRT (« Discourse Representation Theory ») [90] ou, plus récemment, la SDRT [11] ont été adaptées et / ou approximées dans des implémentations informatiques de générateurs du langage naturel, opérant soit à base de règles [50], soit à base de modèles statistiques [108], surtout en situations de monologue ; (ii) une approche basée sur les actes de langage, combinant la théorie traditionnelle des actes de langage de Searle et Vanderveken [156], [185] avec des résultats obtenus avec l'approche BDI à la modélisation des agents virtuels [27], [133] : ainsi, des implémentations informatiques pour la génération en dialogue sont apparues ; ces implémentations se basaient sur diverses taxinomies des actes de langage [177], dérivées de la théorie traditionnelle.

Pour ce qui tient à l'utilisation de la SDRT en génération, les travaux de Danlos [50] et Roussarie [149] peuvent être considérés comme précurseurs ; en ce qui concerne les extensions de la SDRT pour intégrer des aspects liés au dialogue (de toute manière, seulement pour l'interprétation des répliques), les travaux de Prévot et son équipe à Toulouse peuvent être mis en évidence [111]. Finalement, eu égard aux approximations de la SDRT dans des formalismes logiques moins « lourds » (par rapport à la sémantique DRT de base), les travaux de Staudacher [161] peuvent être mentionnées, au sens où ce dernier essaye de remplacer, pour une partie de la SDRT, les représentations DRT des énoncés, avec des représentations dans la logique dynamique

des prédicats¹.

L'originalité de l'approche décrite dans ce chapitre réside, d'une part, dans l'utilisation d'un formalisme logique moins complexe (d'un point de vue computationnel) que les sémantiques dynamiques – la logique des prédicats du premier ordre, où les quantifications sont paramétrées par un ensemble de prédicats discursifs qui spécifient la portée des entités présentes dans les formules logiques qui rendent compte du contenu des énoncés. L'avantage de cette approche est, à un niveau pratique, qu'elle permet d'utiliser des outils et environnements informatiques conçus pour la logique du premier ordre (par exemple, les différentes variantes de Prolog), d'une manière assez immédiate. Ensuite, d'un point de vue méthodologique, l'approche proposée est indépendante de la tâche, car les prédicats de discours ne dépendent pas des contraintes imposées par un domaine d'application particulier ; le couplage avec les aspects qui tiennent à la tâche se fait *via* une ontologie de tâches, manipulée par le contrôleur de tâche (voir chapitre 1) dans un système de dialogue [35], [122]. Cela permet d'avoir plus de portabilité au niveau de la génération des réponses, en diminuant ainsi les coûts d'adaptation à une nouvelle tâche. D'une autre part, ce calcul sémantique est ensuite complété par un filtrage, que nous appelons « pragmatique », où les contraintes induites par les types d'actes de langage sur les énoncés sont pris en compte afin de filtrer l'ensemble des relations rhétoriques qui peuvent relier ces énoncés.

3.2.2 Emulation de la SDRT dans une logique du premier ordre

Position du problème

Pour les buts assumés dans nos recherches (qui est, nous le rappelons, de concevoir et développer un cadre pour le contrôle, sur des bases pragmatiques, de la génération des réponses de la machine en dialogue orienté service, à plusieurs locuteurs), nous nous sommes concentrés sur un ensemble de 17 relations rhétoriques dans la SDRT. Le choix de cet ensemble de relations rhétoriques, des 35 disponibles dans la SDRT d'origine [11] est motivé par des analyses que nous avons effectuées sur des dialogues entre homme et machine (le corpus PVE – « Portail Vocal pour l'Entreprise » [35], [122], et entre plusieurs humains (des pièces de théâtre – voir chapitre 4). Ces 17 relations rhétoriques, censées être utilisées au niveau « pragmatique » pour la structuration rhétorique des énoncés générés par la machine en dialogue, sont :

- des relations **monologiques** – relations spécifiques aux séquences monologiques et utilisées pour relier les énoncés à l'intérieur d'un même tour de parole ; ces relations sont : *Alternation*, *Arrière-plan (Background)*, *Conséquence (Consequence)*, *Elaboration*, *Narration*, *Contraste (Contrast)* et *Parallèle (Parallel)*, et ont les sémantiques informelles telles que définies par Asher et Lascarides [11] ;
- des relations **dialogiques** – *Elaboration de Question (Q-Elab)*, *Paire Question-Réponse (QAP)*, *Paire Question-Réponse Indirecte (IQAP)*, *Correction de Plan (P-Corr)*, *Elaboration de Plan (P-Elab)*, *Background_q*, *Elab_q*, *Narration_q*, *Confirmation (ACK)* et *Pas Assez d'Informations (NEI)* ; les sémantiques informelles de ces relations sont définies par Asher et Lascarides dans la SDRT [11].

¹Cela pourrait être une piste intéressante pour nous-même, pour mieux gérer les phénomènes de correction en dialogue.

Ces relations rhétoriques sont des types d'actes de langage, qui reflètent les dépendances entre le succès de l'acte de langage courant, et le contenu d'un ensemble d'actes précédents. Par conséquent, un dialogue est considéré comme *cohérent* du point de vue pragmatique si et seulement si, pour tout énoncé il existe au moins un lien, *via* une relation de discours, entre ce dernier et un autre acte de langage appartenant à l'historique du dialogue (c'est-à-dire, à l'ensemble des tours de parole précédents en dialogue). Un tel lien est appelé But relatif à un acte de langage (SARG – «Speech Act Related Goal»).

En interprétation pragmatique du langage, les SARG sont récupérés à partir du contenu du discours (et des énoncés), tandis qu'en génération, ces SARG sont définis par les actes de langage *réalisés* par les clauses provenant du contrôleur de dialogue [35]; ils sont donc *a priori* donnés au générateur. Par conséquent, en génération du langage le problème n'est pas d'identifier les SARG, mais plutôt de les *placer* dans un contexte discursif existant.

Nous considérons seulement un type particulier de dialogues, présents dans des situations de réservation de salles auprès d'un secrétariat, emprunt de livres dans une bibliothèque ou, de manière plus générale, dans des situations où le partage collectif des intervalles temporels d'utilisation d'un ensemble précisé de ressources est essentiel.

A l'instar des travaux de Schlangen *et al.* [153], la SDRT est approximée, au sens où les réponses que le système est censé produire sont limitées aux informations concernant : (i) la description des créneaux de temps de disponibilité d'un certain objet (par exemple, livre) ou ressource, en rapport avec un intervalle de temps pour l'utilisateur ; (ii) ou la non-adéquation de cet intervalle de temps convenable pour l'utilisateur. Autrement dit, nous nous intéressons à ce que dans l'intervalle de temps convenable pour l'utilisateur, celui-ci peut avoir accès à l'objet de son désir, ou non.

Par conséquent, le générateur devrait produire des énoncés qui précisent l'intervalle temporel Δt proposé par l'utilisateur (noté par U), en termes d'adéquation aux possibilités – par exemple, d'emprunter un certain livre (`good_time(Δt)`), ou non-adéquation (`bad_time(Δt)`).

Les règles pour calculer les relations de discours qui relient l'intention communicationnelle calculée par le contrôleur de dialogue, au contexte discursif, sont monotones dans notre modèle, tandis que dans la SDRT d'origine, ces règles sont non-monotones. Ce choix est motivé par la possibilité d'éviter des vérifications de cohérence coûteuses en temps de calcul, présentant cependant l'inconvénient d'une relative difficulté à modéliser les dialogues qui contiennent des « divergences » (c'est-à-dire, des remises en cause – des négations de contenu, par exemple, 'Ferme la porte !' / 'Mais elle est déjà fermée !', ou des remises en question – des négations du but, par exemple, 'Ferme la porte !' / 'Non, je ne suis pas ton esclave !').

Le fait de connaître le but général *a priori* des utilisateurs en dialogue (but qui est fourni sous forme logique par le contrôleur de dialogue), en rapport avec les aspects temporels de la conversation, nous permet de formuler les sémantiques des relations rhétoriques du premier ordre.

Les sémantiques de ces relations rhétoriques sont exprimées en termes de prédicats et fonctions de discours ; ces prédicats sont structurés comme un ensemble de règles qui les relient. Nous allons d'abord présenter les définitions des prédicats discursifs, avant de préciser les sémantiques des relations rhétoriques.

Prédicats discursifs

Les prédicats discursifs qui constituent les atomes des sémantiques des relations rhétoriques sont reliés *via* les connecteurs classiques en logique – la conjonction (\wedge), la disjonction (\vee), la négation (\neg), et l'implication (\Rightarrow) ; les arguments de ces prédicats sont en général constitués par les énoncés ou, de manière récursive, par des prédicats appliqués aux énoncés.

Afin d'augmenter l'expressivité des sémantiques des relations rhétoriques, nous avons considéré une taxinomie assez fine de moments de temps, cela parce que les dialogues formalisés par notre cadre concernent surtout la négociation des intervalles temporels. Ainsi, cette taxinomie délimite les moments de temps en : (i) $t_{\#}$ – conditionnel, (ii) t_+ – futur, (iii) t – présent, (iv) t_- – passé, (v) $t_{=}$ – plus que parfait, (vi) t_{\pm} – conditionnel passé, (vii) t_{\mp} – futur antérieur, (viii) t_{\exists} – un moment pas précisément situé sur l'axe du temps, et (ix) t_{\forall} – n'importe quel moment de temps.

Les prédicats discursifs utilisés sont de plusieurs types : d'abord, nous considérons les prédicats **ensemblistes**, qui spécifient les relations entre des ensembles d'entités : (i) \in – MemberOf, (ii) \subset – SubclassOf, (iii) \ni – ClassOf, (iv) \supset – SuperclassOf, (v) $\cap = \emptyset$ – Disjoint, (vi) $\cup = \text{All}$ – ExhaustiveDecomposition, (vii) $(\cap = \emptyset) \wedge (\cup = \text{All})$ – Partition. Ces prédicats sont assez classiques, au sens où on les rencontre, en général, dans n'importe quelle base de connaissances [150]. Il en va de même pour une deuxième catégorie de prédicats, de **mesure** : smaller, greater, et equals².

Un troisième type de prédicats, **discursifs** sont plus étroitement liés aux particularités des dialogues modélisés (par la prise en compte explicite des aspects temporels). Ces prédicats, ainsi que certaines fonctions, toujours reliés aux particularités des discours modélisés, seront présentés ci-dessous.

Ainsi, en désignant par $K/1$ la fonction qui, prenant comme argument l'étiquette d'un énoncé retourne son expression logique, et par $t/1$ la fonction qui, prenant comme argument l'étiquette d'un énoncé retourne le moment dans le temps où l'énoncé est produit, nous pouvons exprimer de manière compositionnelle les sémantiques des prédicats discursifs.

Le premier prédicat, $\text{question}/1$, est vrai si son argument est une question, ce qui revient à ce que sa forme logique contienne des variables non-initialisées ; au niveau sémantique, cela se formalise comme suit :

$$\text{question}(\alpha) ::= \exists v : \text{MemberOf}(v, K(\alpha)) \wedge \neg \exists \omega : \text{MemberOf}(\omega, \Omega) \wedge \text{equals}(v, \omega).$$

Une définition similaire est donnée pour le prédicat $\text{enounce}/1$, vrai si son argument n'est pas une question. Ainsi :

$$\text{enounce}(\alpha) ::= \neg \text{question}(\alpha).$$

Le prédicat $\text{confirmation}/2$ est vrai si l'énoncé qui apparaît en tant que son deuxième argument confirme l'énoncé qui apparaît en tant que son premier argument, et les deux énoncés ont été produits par des locuteurs différents. Nous observons donc que ce prédicat est une version plus souple de la relation rhétorique de Confirmation (*ACK*). Ainsi,

$$\text{confirmation}(\alpha, \beta) ::= \text{equals}(K(\beta), K(\alpha)) \wedge \neg \text{equals}(\text{emitter}(\beta), \text{emitter}(\alpha)).$$

La fonction $\text{answer}/2$ retourne l'ensemble des formes logiques qui représentent des réponses

²La définition de ce prédicat est toutefois plus subtile, parce qu'en fait nous regroupons sous le même nom des comportements différents – voir plus loin.

à l'énoncé pris comme argument. En termes formels,

$$\text{answer}(\beta, \alpha) ::= \text{equals}(\text{topic}(\beta), \text{topic}(\alpha)) \wedge \text{greater}(t(\beta), t(\alpha)) \wedge \forall v : \text{MemberOf}(v, K(\beta)) \exists \omega : \text{MemberOf}(\omega, \Omega) \wedge \text{equals}(v, \omega).$$

La fonction `topic/1` mérite plus de discussion, car elle essaye de remplacer la relation de discours de Topique (\Downarrow) dans la SDRT d'origine. Notre fonction ne fait que retourner une structure qui contient les types et valeurs des variables initialisées dans l'énoncé qu'elle prend comme argument. Donc, dans notre émulation de la SDRT le topique est une caractéristique de l'énoncé, au lieu d'être une relation entre une paire d'énoncés ou de constituants discursifs, comme dans la SDRT d'origine. Cela nous permet, à notre avis, plus de flexibilité dans la manipulation des topiques relatifs à des paires d'énoncés (on peut composer de diverses façons les valeurs de cette fonction). Au niveau formel, cette fonction s'exprime ainsi :

$$\text{topic}(\alpha) ::= \text{ExhaustiveDecomposition}(v, \omega) \wedge \text{MemberOf}(v, K(\alpha)) \wedge \text{MemberOf}(\omega, \Omega) \wedge \text{equals}(v, \omega).$$

La fonction `emitter/1` retourne l'identifiant du locuteur qui a produit l'énoncé pris comme argument ; ce locuteur peut être soit M (la machine), soit U (*un* utilisateur humain). Formellement :

$$\text{emitter}(\alpha) ::= (U \wedge \neg \text{Disjoint}(U, K(\alpha))) \vee (M \wedge \neg \text{Disjoint}(M, K(\alpha))).$$

Le prédicat `entails/2` est vrai si l'énoncé pris comme premier argument implique l'énoncé pris comme deuxième argument, et le premier énoncé est vrai (du point de vue de son émetteur), à tout moment dans le temps. Formellement :

$$\text{entails}(\alpha, \beta) ::= (K(\alpha) \Rightarrow K(\beta)) \wedge \text{equals}(\Delta t(\alpha), t_v).$$

La fonction `SARG/1` retourne le but en relation avec l'acte de langage de l'énoncé (le « Speech act related goal ») de l'énoncé pris comme argument [11]. Cependant, dans notre cadre, cette fonction retourne tout simplement le type de l'acte de langage réalisé par l'énoncé. Formellement³ :

$$\text{SARG}(\alpha) ::= \chi : (\text{SubclassOf}(\chi, K(\alpha)) \vee \text{equals}(\chi, K(\alpha))) \wedge \exists \gamma : \text{MemberOf}(\gamma, \Gamma) \wedge \neg \text{Disjoint}(\gamma, \chi).$$

Les prédicats `good_time/1` et `bad_time/1` ont les mêmes significations que dans le projet Verbmobil [153] : ils sont vrais si leur argument est un moment dans le temps (spécifié dans un énoncé courant), qui correspond à un autre moment dans le temps, dans un énoncé antérieur, qui partage (partiellement) le topique avec le premier énoncé (il s'agit de l'énoncé courant). Formellement,

$$\text{good_time}(\Delta t(\beta)) ::= \exists \alpha : \neg \text{Disjoint}(\text{topic}(\alpha), \text{topic}(\beta)) \wedge \text{smaller}(t(\alpha), t(\beta)) \wedge (\text{SubclassOf}(\Delta t(\beta), \Delta t(\alpha)) \vee \text{equals}(\Delta t(\alpha), \Delta t(\beta)));$$

$$\text{bad_time}(\Delta t(\beta)) ::= \neg \text{good_time}(\Delta t(\beta)).$$

Les fonctions `$\Delta t/1$` et `$t/1$` méritent une discussion particulière, car, quoiqu'elles soient toutes les deux des fonctions qui prennent des énoncés comme arguments et retournent des moments dans le temps, la première fonction retourne l'intervalle de temps proposé dans le premier énoncé (ou, où l'événement dont il s'agit dans cet énoncé se passe), tandis que la deuxième fonction retourne tout simplement le moment dans le temps (relatif au début du dialogue) quand cet énoncé est produit. Ainsi, tandis que la fonction `t` est calculée par une simple gestion interne des

³L'ensemble Γ contient les cinq types d'acte de langage utilisés dans cette thèse : « faire-faire », « faire-savoir », « faire-faire-savoir », « faire-pouvoir » et « faire-devoir » – voir plus loin dans ce chapitre.

énoncés dans le contrôleur de dialogue, la fonction Δt est calculée en vertu d'une sémantique reliée à l'énoncé auquel la fonction est appliquée. Ainsi :

$$\Delta t(\beta) ::= (t_2 - t_1) : \exists t_1, t_2 : (\text{MemberOf}(t_2 - t_1, K(\beta))) \vee (\text{MemberOf}(t_2 - t_1, \text{topic}(\beta)) \wedge \text{MemberOf}(t_1, K(\beta)) \wedge \text{MemberOf}(t_2, K(\beta))).$$

Toutes ces spécifications nous permettront de construire, de manière relativement simple, un algorithme pour la mise à jour de la structure rhétorique en dialogue, à partir des sémantiques des énoncés et des relations rhétoriques. Le lien entre les énoncés et les sémantiques des relations rhétoriques est constitué justement par ces prédicats et fonctions discursifs, comme nous le verrons plus loin dans ce chapitre.

Sémantiques des relations rhétoriques

Une fois définis les prédicats discursifs, nous pouvons préciser les sémantiques des 17 relations rhétoriques retenues. En ce qui concerne les relations monologiques, nous en avons considéré sept :

▷ l'*Alternation*, une relation coordonnante qui relie deux énoncés tels qu'un est l'alternative de l'autre (cela correspond au « OU » logique). Exemple :

M : Nous pouvons soit vous prêter le livre 'Retour en Amérique' de Michel Ardan^α,
M : soit vous donner le DVD 'Sur les traces des peaux rouges', sur le même sujet.^β

La sémantique de la relation d'*Alternation* est :

$$\text{Alternation}(\alpha, \beta) ::= \text{enounce}(\alpha) \wedge \text{enounce}(\beta) \wedge (K(\alpha) \vee K(\beta)).$$

▷ la relation d'Arrière-plan (*Background*) ; il s'agit d'une relation coordonnante qui relie deux énoncés ou constituants discursifs tels que le deuxième précise un état temporellement antérieur à l'événement précisé dans le premier énoncé. Exemple :

M : Je ne peux pas vous prêter le livre 'Retour en Amérique' de Michel Ardan.^α
M : Vous avez le droit d'emprunter trois livres au maximum.^β

La sémantique de la relation de *Background* est :

$$\text{Background}(\alpha, \beta) ::= \text{enounce}(\alpha) \wedge \text{enounce}(\beta) \wedge \text{entails}(\alpha, \beta).$$

▷ la Conséquence (*Consequence*) ; il s'agit d'une relation coordonnante qui correspond à l'implication logique dynamique (\Rightarrow), sauf que les deux énoncés n'ont pas forcément besoin d'être vrais dans tout modèle⁴. Dans notre cadre, nous allons simuler cette relation par une implication statique *locale* (c'est-à-dire, à valeur limitée au point de vue du locuteur qui a produit les deux énoncés). Exemple :

M : Si on vous prête le livre 'Retour en Amérique' de Michel Ardan^α,
M : alors dans la bibliothèque il n'y aura plus aucun exemplaire de ce livre !^β

⁴Pour une discussion sur ce sujet, voir plus loin dans ce chapitre.

La sémantique de *Consequence* est :

$$\text{Consequence}(\alpha, \beta) ::= \text{enounce}(\alpha) \wedge \text{enounce}(\beta) \wedge (K(\alpha) \Rightarrow K(\beta)).$$

▷ l'*Elaboration* ; il s'agit d'une relation subordonnante qui relie deux énoncés ou constituants discursifs tels que le deuxième élabore sur le contenu du premier, et les deux énoncés partagent le même topique⁵. Exemple :

M : Je ne peux pas vous prêter le livre 'Retour en Amérique' de Michel Ardan !^α

M : Ceci est le seul exemplaire dans la bibliothèque !^β

La sémantique de l'*Elaboration* est :

$$\text{Elaboration}(\alpha, \beta) ::= \text{enounce}(\alpha) \wedge \text{enounce}(\beta) \wedge (\text{SubclassOf}(\text{topic}(\beta), \text{topic}(\alpha)) \vee \text{equals}(\text{topic}(\alpha), \text{topic}(\beta))).$$

▷ la *Narration* ; il s'agit d'une relation coordonnante qui relie deux énoncés qui partagent le même topique et dont les événements dont il s'agit dans ces énoncés sont consécutifs dans la temps. Exemple :

M : Vous allez sur le hall intitulé « Littérature Scandinave »^α.

M : Ensuite, vous cherchez la rangée dénommée « Andersen »^β.

La sémantique de la *Narration* est :

$$\text{Narration}(\alpha, \beta) ::= \text{enounce}(\alpha) \wedge \text{enounce}(\beta) \wedge \neg \text{Disjoint}(\text{topic}(\alpha), \text{topic}(\beta)) \wedge \text{smaller}(\Delta t(\alpha), \Delta t(\beta)).$$

▷ le *Contraste* (*Contrast*) ; il s'agit d'une relation coordonnante qui relie deux énoncés ou constituants discursifs avec des sémantiques identiques, à l'exception d'un ou quelques éléments (qui sont opposés – c'est-à-dire, un est ou implique la négation de l'autre). Exemple :

M : Je peux vous prêter le livre 'Retour en Amérique' de Michel Ardan^α.

M : Par contre, le livre 'Voyage en Espagne' de Gustave Flaubert est disponible seulement en consultation sur place^β.

La sémantique de *Contrast* est :

$$\text{Contrast}(\alpha, \beta) ::= \text{enounce}(\alpha) \wedge \text{enounce}(\beta) \wedge \text{equals}(\text{size}(K(\alpha) \setminus K(\beta)), 1) \wedge \text{entails}(\alpha, \neg\beta).$$

▷ la relation de *Parallèle* (*Parallel*) ; il s'agit toujours d'une relation coordonnante, qui relie deux énoncés ou constituants discursifs qui ont une structure sémantique similaire (c'est-à-dire, différentes seulement par les valeurs de quelques variables, mais sans que cela entraîne une opposition au niveau sémantique), des temps identiques pour les événements décrits, et dont les topiques sont non-disjoints. Exemple :

M : Je peux vous prêter le livre 'Retour en Amérique' de Michel Ardan^α.

M : Je peux aussi vous montrer le DVD 'Sur les traces des peaux rouges' sur le même sujet^β.

⁵Pour une discussion sur la manière dont nous formalisons la notion de topique, voir plus loin dans ce chapitre.

La sémantique de *Parallel* est :

$$\text{Parallel}(\alpha, \beta) ::= \text{enounce}(\alpha) \wedge \text{enounce}(\beta) \wedge \neg \text{Disjoint}(\text{topic}(\alpha), \text{topic}(\beta)) \wedge \neg \text{Disjoint}(K(\alpha), K(\beta)) \wedge \text{equals}(\Delta t(\alpha), \Delta t(\beta)).$$

En ce qui concerne les relations rhétoriques dialogiques, nous en avons retenu dix :

► l'Elaboration de Question *Q-Elab*, qui est une relation *subordonnante* (l'énoncé introduit par cette relation est subordonné, du point de vue thématique, à l'énoncé auquel ce dernier est relié), connecte deux énoncés ou constituants discursifs (c'est-à-dire, ensembles d'énoncés, reliés par des relations rhétoriques ; ces constituants discursifs se définissent de manière récursive sur l'ensemble des énoncés) tels que le deuxième énoncé est une question à laquelle toute réponse élabore un plan pour atteindre un des buts posés dans le premier énoncé (ou constituant discursif). Pour aider la compréhension, nous considérons un exemple, où α et β désignent deux énoncés, U et M désignent respectivement un utilisateur et la machine, et le domaine concerné est la réservation de livres dans une bibliothèque :

U : Je voudrais lire ce bouquin lundi prochain ^{α} .

M : A 14h, ça vous ira ? ^{β}

L'expression du premier ordre qui émule la sémantique de *Q-Elab* est présentée comme une clause de Horn, où les significations des connecteurs logiques \wedge , \vee , \neg et \Rightarrow , et des quantificateurs \forall et \exists sont usuelles [63] :

$$\text{Q-Elab}(\alpha, \beta) ::= \text{enounce}(\alpha) \wedge \text{question}(\beta) \wedge \neg \text{Disjoint}(\Delta t(\beta), SARG(\alpha)) \wedge (\forall \chi : \text{answer}(\chi, \beta) \neg \Rightarrow \neg SARG(\alpha)).$$

► la Paire Question-Réponse (*QAP*) ; il s'agit d'une relation *subordonnante* qui relie deux énoncés ou constituants discursifs tels que le premier énoncé est une question et le deuxième, une réponse directe à cette question⁶. Exemple :

U : Où pourrais-je trouver le bouquin ayant la cote 'D01211108' ? ^{α}

M : Au deuxième étage, à droite, sur la deuxième rangée ^{β} .

La sémantique de *QAP* est :

$$\text{QAP}(\alpha, \beta) ::= \text{question}(\alpha) \wedge \text{enounce}(\beta) \wedge \neg \text{Disjoint}(\text{topic}(\alpha), \text{topic}(\beta)) \wedge \text{answer}(\beta, \alpha).$$

► la Paire Question-Réponse Indirecte (*IQAP*), qui est une relation *subordonnante*, relie deux énoncés ou constituants discursifs, tels que le premier énoncé est une question, et le deuxième est une réponse qui fournit des informations à partir desquelles l'émetteur de la question peut inférer une réponse à cette question. En gardant les mêmes conventions de notation que ci-dessus, nous présentons un exemple :

U : Est-ce que je pourrais avoir ce livre pour cette semaine ? ^{α}

M : Le livre revient dans notre bibliothèque lundi prochain ^{β} .

⁶Ici, par « réponse directe » nous entendons une réponse qui permet d'initialiser seulement les variables qui n'étaient pas initialisées dans la forme logique de la question, et cela sans offrir plus d'informations, non demandées dans la question.

La sémantique du premier ordre de *IQAP* est :

$$IQAP(\alpha, \beta) ::= \text{question}(\alpha) \wedge \text{enonce}(\beta) \wedge (K(\beta) \Rightarrow \neg \text{Disjoint}(\Delta t(\beta), SARG(\alpha))).$$

► la Correction de Plan (*P-Corr*), qui est une relation subordonnante, relie deux énoncés ou constituants discursifs tels que le second exprime le fait que son émetteur réfute le plan de l'allocutaire (plan qui est exprimé dans le premier énoncé) :

U : Est-ce que je pourrais avoir ce livre pour la semaine prochaine ? ^{α}

M : Le livre a déjà été réservé par un autre client. ^{β}

La sémantique de *P-Corr* est :

$$P - Corr(\alpha, \beta) ::= (K(\beta) \Rightarrow \neg SARG(\alpha)) \Rightarrow (\text{bad_time}(\Delta t(\beta)) \wedge \text{SuperclassOf}(\Delta t(\beta), SARG(\alpha))).$$

► l'Elaboration de Plan (*P-Elab*) est une relation subordonnante qui relie deux énoncés ou constituants discursifs tels que le deuxième élabore un plan pour atteindre le but précisé dans le premier énoncé. Par exemple :

U : Est-ce que je pourrais avoir ce livre pour la semaine prochaine ? ^{α}

M : Pour l'avoir, il faut que vous alliez à notre siège 4 rue de la Sapristie. ^{β}

La sémantique de *P-Elab* est :

$$P - Elab(\alpha, \beta) ::= \text{good_time}(\Delta t(\beta)) \wedge \neg \text{Disjoint}(\Delta t(\beta), SARG(\alpha)) \vee \text{bad_time}(\Delta t(\beta)) \wedge \neg \text{equals}(SARG(\alpha) \setminus \Delta t(\beta), \emptyset).$$

► la « Question déterminante d'arrière-plan » *Background_q*, relie deux énoncés ou constituants discursifs tels que le deuxième énoncé est une question dont toute réponse est en relation d'Arrière-plan (*Background* – voir ci-dessous) avec le premier énoncé. Exemple :

M : Ce livre a déjà été emprunté hier par un autre client. ^{α}

U : Est-ce qu'il y a d'autres personnes qui cherchent aussi ce livre ? ^{β}

La sémantique de la relation *Background_q* est :

$$Background_q(\alpha, \beta) ::= \text{enonce}(\alpha) \wedge \text{question}(\beta) \wedge (\forall \chi : \text{answer}(\chi, \beta) \Rightarrow Background(\chi, \alpha)).$$

► la « Question déterminante d'élaboration » *Elab_q* est une relation subordonnante qui relie deux énoncés ou constituants discursifs tels que le deuxième est une question dont toute réponse est en relation d'Elaboration (*Elab* – voir ci-dessous) avec le premier énoncé. Exemple :

M : Nous avons reçu de nouveaux livres sur votre sujet d'intérêt. ^{α}

U : Pourriez-vous me donner quelques titres ? ^{β}

La sémantique de *Elab_q* est :

$$Elab_q ::= \text{enonce}(\alpha) \wedge \text{question}(\beta) \wedge (\forall \chi : \text{answer}(\chi, \beta) \Rightarrow Elab(\alpha, \chi)).$$

► la « Question déterminante de narration » (*Narration_q*) est une relation subordonnante qui relie deux énoncés ou constituants discursifs tels que le deuxième énoncé est une question dont

toute réponse est en relation de *Narration* (voir ci-dessous) avec le premier énoncé. Un exemple est montré ci-dessous :

M : Vous pourriez commencer avec le livre ‘Narcisse et Goldmund’ de Hesse ^{α} .

U : Et ensuite, qu’est-ce que vous me recommandez ? ^{β}

La sémantique de *Narration_q* est :

$Narration_q ::= \text{enounce}(\alpha) \wedge \text{question}(\beta) \wedge (\forall \chi : \text{answer}(\chi, \beta) \Rightarrow Narration(\alpha, \chi))$.

► la Confirmation (*ACK*) est une relation subordonnante aussi, qui relie deux énoncés ou constituants discursifs, tels que le deuxième énoncé est une confirmation directe (par un locuteur différent de l’émetteur du premier énoncé) du premier énoncé. Exemple :

M : Ce livre vous ira ? ^{α}

U : Oui ^{β} .

La sémantique de *ACK* est :

$ACK(\alpha, \beta) ::= (\text{question}(\alpha) \vee \text{enounce}(\alpha)) \wedge \text{enounce}(\beta) \wedge \neg \text{Disjoint}(\text{topic}(\alpha), \text{topic}(\beta)) \wedge \text{confirmation}(SARG(\alpha), \beta)$.

► la relation de Manque d’information (*NEI* – « Not Enough Information ») est une relation subordonnante qui relie deux énoncés ainsi que le premier est une question, et le deuxième est un énoncé qui affirme le fait que, selon son émetteur, la question ne contient pas assez d’informations pour permettre d’en calculer une réponse. Exemple est :

M : Quel auteur aimeriez vous consulter sur ce sujet ? ^{α}

U : Je ne sais pas ce que vous en avez... ^{β}

La sémantique de *NEI* est :

$NEI(\alpha, \beta) ::= \text{question}(\alpha) \wedge \text{enounce}(\beta) \wedge (K(\beta) \Rightarrow \neg \exists \chi : \text{equals}(\text{emitter}(\chi), \text{emitter}(\beta)) \wedge \text{answer}(\chi, \alpha))$.

Avant de passer à l’algorithme de calcul des relations rhétoriques, nous faisons quelques précisions sur l’ensemble de ces relations. Tout d’abord, pour nous la distinction, présente dans la SDRT d’origine, entre relations véridiques et relations non-véridiques, ne nous semble pas assez pertinente, du point de vue de la génération des énoncés, sous contraintes rhétoriques. Cela parce que nous considérons que toute relation rhétorique est véridique selon le point de vue de l’émetteur de l’énoncé qui est relié aux énoncés antérieurs *via* cette relation, et *a priori* non-véridique pour les autres interlocuteurs. *A priori* non-véridique veut dire qu’une relation rhétorique ne devient véridique pour l’allocutaire que si celui-ci ne met pas en question (de manière explicite, c’est-à-dire, *via* une relation de contradiction, ou implicite, c’est-à-dire, *via* un nouvel énoncé, ou un nouveau topique, inconsistant avec l’acte mis en question) les énoncés reliés par cette relation rhétorique. Autrement dit, au lieu de regarder la vérité d’un énoncé selon une infinité de modèles (mondes possibles), nous rapportons cette vérité au « monde » de chaque locuteur [35].

3.2.3 Mise à jour de la structure de discours dialogique : algorithme

Dans cette section nous allons présenter une procédure pour mettre à jour la structure de discours (SDRS) d'un dialogue, en y ajoutant un énoncé⁷. Plus précisément, pour une formule logique $K(\alpha)$ provenant du contrôleur de dialogue, et qui exprime une intention communicationnelle que nous devons mettre sous forme d'un énoncé linguistique α , nous souhaitons trouver l'ensemble \mathfrak{R} des relations rhétoriques qui relient l'énoncé α à un ensemble \mathfrak{S} d'énoncés antérieurs en dialogue.

L'algorithme pour la mise à jour de la SDRS dialogique prend en entrée une forme logique (pour l'énoncé à ajouter à la structure rhétorique), une structure rhétorique existante, avec les formules logiques de tous les énoncés qui composent cette dernière ; la procédure consiste à parcourir les étapes suivantes :

1. **pour** chaque énoncé, libellé α , que nous devons ajouter à la SDRS dialogique, notée $\bar{\mathfrak{S}}$:
 - (a) lire la forme logique $K(\alpha)$, via une requête au contrôleur de dialogue⁸ ;
 - (b) initialiser les variables : $\mathfrak{S}(\alpha) \leftarrow \emptyset$, $\mathfrak{R}(\alpha) \leftarrow \emptyset$;
 - (c) **pour** chaque énoncé β_i dans $\bar{\mathfrak{S}}$ ($i = 1, \dots, |\bar{\mathfrak{S}}|$) :
 - i. **pour** chaque relation rhétorique ρ_j connue par le système ($j = 1, \dots, 17$) :
 - A. lire les formules logiques $K(\beta_i)$ et Σ_j ;
 - B. calculer la valeur de vérité de $\Sigma_j(\sigma(K(\alpha), K(\beta_i)))$ et la désigner par γ ;
 - C. **si** $\gamma = \text{FALSE}$, **alors aller** à 1.(c).i. ;
sinon, $\mathfrak{R} \leftarrow \mathfrak{R} \cup \{\rho_j\}$ et $\mathfrak{S} \leftarrow \mathfrak{S} \cup \{\beta_i\}$ et **aller** à 1.(c).i. ;
 - D. $\mathfrak{S}(\alpha) \leftarrow \mathfrak{S}(\alpha) \cup \mathfrak{S}$ et $\mathfrak{R}(\alpha) \leftarrow \mathfrak{R}(\alpha) \cup \mathfrak{R}$;
2. calculer la valeur de vérité de $\text{equals}(|\mathfrak{S}(\alpha)|, 0) \wedge \text{equals}(|\mathfrak{R}(\alpha)|, 0)$ et la noter par v :
 - (a) **si** $v = \text{FALSE}$, **alors** :
 - i. ajouter α aux énoncés dans la SDRS courante ;
 - ii. ajouter $\mathfrak{R}(\alpha)$ à l'ensemble des relations rhétoriques dans la SDRS courante ;
 - (b) **sinon**, construire une nouvelle SDRS qui contient seulement un énoncé, α , sans relations rhétoriques.

Dans l'algorithme précisé ci-dessus, nous avons noté par $\bar{\mathfrak{S}}$ l'ensemble de *tous* les énoncés précédant α dans le dialogue courant et nous avons cherché, pour chaque énoncé β_i dans l'historique du dialogue, les relations rhétoriques ρ telles que $\Sigma_\rho(\sigma(K(\alpha), K(\beta_i))) \neq \text{FALSE}$, où

⁷Un énoncé est ajouté à une SDRS si et seulement si au moins une relation rhétorique est trouvée entre ce dernier et les énoncés dans l'historique du dialogue ; pour le moment nous ne prenons donc pas en compte des contraintes supplémentaires sur l'ensemble des énoncés antérieurs accessibles, comme la contrainte de « frontière droite » [11].

⁸En fait, tout d'abord le contrôleur de dialogue signale au générateur qu'il y a une nouvelle formule logique à générer sous forme linguistique, en lui envoyant l'étiquette de cet énoncé (en fait, un pointeur vers la forme logique correspondante) ; ensuite, le générateur, lorsqu'il est prêt à traiter cette nouvelle forme logique, signale cela au contrôleur de dialogue, en lui envoyant une requête pour cette forme logique.

σ désigne l'opération de permutation⁹, Σ_ρ désigne la sémantique formelle de la relation ρ , $|X|$ désigne la cardinalité de l'ensemble X et \cup , la réunion d'ensembles.

Cet algorithme construit une structure de discours *complète*, au sens où toutes les relations rhétoriques logiquement possibles entre des paires d'énoncés (dont au moins un est censé être généré par la machine) sont calculées ; les contradictions possibles impliquées par cette approche (déterminées par le fait qu'une relation rhétorique peut être révisée par la présence d'une relation ultérieure, ou par des énoncés ultérieurs) sont contournées par le caractère monotone du cadre logique utilisé. Cela conduit au calcul de toutes les structures rhétoriques potentielles, sans en trouver une qui soit la plus cohérente. Par conséquent, des processus supplémentaires de filtrage sur ces structures seront appliquées, afin de renforcer le rôle du contexte (dialogique) pour ainsi déterminer la structure rhétorique la plus cohérente (au sens de Asher [11], où plus le nombre des relations rhétoriques qui relient un ensemble donné d'énoncés est grand, et plus le nombre des sous-spécifications – c'est-à-dire, en fin de compte, les variables non-initialisées – est important, plus la structure rhétorique de cet ensemble d'énoncés est cohérente). Cette prise en compte du contexte afin de ne plus obtenir qu'une seule structure rhétorique, sera décrite dans le chapitre 4.

3.2.4 Calcul des relations rhétoriques : exemple étendu

Dans l'algorithme présenté ci-dessus, l'étape essentielle est 1.(c).i.B ; cette étape sera détaillée *via* un exemple dans cette section. Plus précisément, nous allons montrer comment les valeurs de vérité des propositions $\Sigma_j(\sigma(K(\alpha), K(\beta_i)))$ sont calculées, où i indexe l'ensemble des énoncés, disponibles sous forme de clauses logiques dans la structure de discours, et j indexe l'ensemble des relations rhétoriques données (dans notre cas donc, $j = 1, \dots, 17$).

Pour ce faire, nous considérons un fragment de dialogue artificiel, consistant en deux répliques, une d'un usager, l'autre de la machine ; le dialogue concerne toujours le domaine de la réservation de livres dans une bibliothèque :

U : Je prendrai ce livre lundi ^{α} .

M : Est-ce bien à 14 h ? ^{β}

En fait, l'énoncé α provient de l'usager, donc il est disponible en même temps sous forme textuelle et sous forme logique¹⁰. En revanche, l'énoncé β n'est disponible que sous forme logique, puisqu'il exprime une intention communicationnelle, produite par le contrôleur de dialogue [35] ; sa forme textuelle doit être déterminée par le générateur de langage (par exemple *via* un choix, piloté par le contexte, dans un ensemble de structures figées). Plus précisément, il s'agit de calculer les relations rhétoriques qui peuvent relier les énoncés α et β entre eux. Pour cela, en appliquant l'algorithme présenté ci-dessus, nous essayons chacune des dix relations rhétoriques dialogiques (on peut ainsi limiter l'espace de recherche aux relations dialogiques, parce que nous pouvons avoir facilement accès au fait que les deux énoncés n'ont pas été produits par le même locuteur). Suite à ce processus, la relation *Q-Elab* est trouvée entre ces deux énoncés.

⁹On voit donc que, à la différence de la SDRT d'origine et des discussions de Asher [12] ou Scherrer [155], nous ne prenons pas en compte l'ordre des arguments des relations rhétoriques. Nous considérons toutefois cela comme important, pour raffiner ces structures rhétoriques.

¹⁰La formule logique pour l'énoncé produit par U est calculée par l'analyseur sémantique dans l'architecture de dialogue (voir chapitre 1).

Afin d'illustrer en détail le déroulement de l'algorithme, nous supposons que c'est précisément la relation $Q\text{-Elab}(\alpha, \beta)$ qui est vérifiée :

1. nous trouvons les clauses qui expriment les sémantiques des énoncés α et β :
 - $\alpha \mapsto K(\alpha) ::= \exists X, Y : \text{object}(X) \wedge \text{equals}(X, \text{book}) \wedge \text{agent}(Y) \wedge \text{equals}(Y, U) \wedge \text{equals}(t(\alpha), t) \wedge \text{read}(Y, X) \wedge \text{equals}(\Delta t(\alpha), t_+)$;
 - $\beta \mapsto K(\beta) ::= \exists X, Y : \text{object}(X) \wedge \text{equals}(X, \text{book}) \wedge \text{agent}(Y) \wedge \text{equals}(Y, U) \wedge \text{read}(Y, X) \wedge \text{greater}(t(\beta), t(\alpha)) \wedge \text{equals}(\Delta t(\beta), '14h') \wedge \text{equals}(\Delta t(\beta), t_+) \wedge \text{equals}(\text{good_time}(\Delta t(\beta)), ?)$;
2. nous récupérons la sémantique de la relation rhétorique courante :

$$\Sigma_{Q\text{-Elab}} ::= \text{enounce}(\alpha) \wedge \text{question}(\beta) \wedge \neg \text{Disjoint}(\Delta t(\beta), SARG(\alpha)) \wedge (\forall \gamma : \text{answer}(\gamma, \beta) K(\gamma) \Rightarrow \neg SARG(\alpha)) ;$$
3. nous éclatons chaque entité dans la sémantique de cette relation rhétorique :
 - $\text{enounce}(\alpha) ::= \neg \text{question}(\alpha) \mapsto \forall v : \text{MemberOf}(v, K(\alpha)) \vee \exists \omega : \text{MemberOf}(\omega, \Omega) \vee \text{equals}(v, \omega)$;
 - $\text{question}(\beta) ::= \exists v' : \text{MemberOf}(v', K(\beta)) \wedge \neg \exists \omega' : \text{MemberOf}(\omega', \Omega) \wedge \text{equals}(v', \omega')$;
 - $SARG(\alpha) ::= \exists X, Y, \Theta : \text{object}(X) \wedge \text{equals}(X, 'book') \wedge \text{agent}(Y) \wedge \text{equals}(Y, U) \wedge \text{good_time}(\Theta) \wedge \text{equals}(\Theta, \Delta t(\alpha)) \wedge \text{greater}(\Theta, 'lundi')$;
 - $\Delta t(\beta) ::= '14h' \wedge 'lundi'$;
 - $\forall \delta : \text{answer}(\delta, \beta) ::= \forall \delta : \text{greater}(t(\delta), t(\beta)) \wedge \text{equals}(\text{topic}(\delta), \text{topic}(\beta)) \wedge \forall v'' : \text{MemberOf}(v'', K(\delta)) \Rightarrow \exists \omega'' : \text{MemberOf}(\omega'', \Omega) \wedge \text{equals}(v'', \omega'')$;
 - $\neg SARG(\alpha) ::= \forall X, Y, \Theta : \text{object}(X) \vee \text{equals}(X, 'book') \vee \text{agent}(Y) \vee \text{equals}(Y, U) \vee \text{bad_time}(\Theta) \vee \text{equals}(\Theta, \Delta t(\alpha)) \vee \text{greater}(\Theta, 'lundi')$;
 - $\text{topic}(\beta) ::= \text{ExhaustiveDecomposition}(\text{book}, \text{read}, \text{good_time}('14h'), \text{good_time}('lundi'), t_+)$;
 - $\text{good_time}(\Theta) ::= \exists \gamma, \pi : \neg \text{Disjoint}(\text{topic}(\gamma), \text{topic}(\pi)) \wedge \text{smaller}(t(\alpha), t(\pi)) \wedge (\text{SubclassOf}(\Theta, \Delta t(\alpha)) \vee \text{equals}(\Theta, \Delta t(\alpha))) \wedge \pi : \text{equals}(\Delta t(\pi), \Theta)$;
4. nous calculons la valeur de vérité de chaque clause entre les conjonctions, dans la sémantique de la relation rhétorique courante :
 - $\text{enounce}(\alpha) \wedge \text{question}(\beta) \rightarrow \text{TRUE}$; cette expression est obtenue en substituant la sémantique de α dans la sémantique de la clause montrée ci-dessus, et en explorant l'ensemble des prédicats discursifs *et* dans l'ontologie de tâches (notée par Ω) toutes les valeurs possibles pour les variables présentes ;
 - $\neg \text{Disjoint}(\Delta t(\beta), SARG(\alpha)) \rightarrow \text{TRUE}$; cela résulte directement de la clause : $\text{SubclassOf}(\Delta t(\beta), \Delta t(\alpha)) \wedge \text{SubclassOf}(\Delta t(\alpha), SARG(\alpha))$;
 - $\forall \delta : \text{answer}(\delta, \beta) \neg \Rightarrow \neg SARG(\alpha) ::= \forall \delta : \text{greater}(t(\delta), t(\beta)) \wedge \text{equals}(\text{topic}(\delta), \text{ExhaustiveDecomposition}('book', 'read', \text{good_time}('14h'), \text{good_time}('lundi'), t_+)) \wedge \forall v'' : \text{MemberOf}(v'', K(\delta)) \Rightarrow \exists \omega'' : \text{MemberOf}(\omega'', \Omega) \wedge \text{equals}(v'', \omega'') \neg \Rightarrow \forall X, Y : \text{object}(X) \vee \text{equals}(X, 'book') \vee \text{agent}(Y) \vee \text{equals}(Y, U) \vee \text{bad_time}(\Delta t(\beta)) \vee \text{equals}(\Delta t(\beta), \Delta t(\alpha)) \vee \text{greater}(\Delta t(\beta), 'lundi')$;
la dernière formule a été obtenue en substituant les prédicats et les fonctions par leurs définitions explicites, et la variable Θ , par $\Delta t(\beta)$; ensuite, par la *skolémisation* (c'est-à-dire, élimination des quantificateurs existentiels) des variables dans cette formule, nous obtenons :

$\forall \delta : \text{answer}(\delta, \beta) \neg \Rightarrow \neg \text{SARG}(\alpha) ::= \forall \delta : \text{greater}(t(\delta), t(\beta)) \wedge \text{equals}(\text{topic}(\delta),$
 $\text{ExhaustiveDecomposition}(\text{'book'}, \text{'read'}, \text{good_time}(\text{'14h'}), \text{good_time}(\text{'lundi'}),$
 $t_+)) \wedge \forall v'' : \text{MemberOf}(v'', K(\delta)) \Rightarrow \text{MemberOf}(\omega_0, \Omega) \wedge \text{equals}(v'', \omega_0) \neg \Rightarrow \forall X, Y :$
 $\text{object}(X) \vee \text{equals}(X, \text{'book'}) \vee \text{agent}(Y) \vee \text{equals}(Y, U) \vee \text{bad_time}(\Delta t(\beta)) \vee$
 $\text{equals}(\Delta t(\beta), \Delta t(\alpha)) \vee \text{greater}(\Delta t(\beta), \text{'lundi'}) ;$
 ensuite, nous éliminons les quantificateurs universels et nous explorons les chemins
 possibles dans l'ontologie de tâches, en obtenant dans notre exemple la valeur de TRUE.

Cet exemple montre qu'il est possible, en principe, de calculer une relation rhétorique reliant un énoncé courant (censé être généré sous forme linguistique), disponible seulement sous forme logique, à un dialogue en déroulement, cela tout en utilisant une logique du premier ordre et un ensemble de prédicats discursifs indépendants de la tâche.

3.3 Filtrage pragmatique des relations rhétoriques

3.3.1 Actes de langage et calcul des relations rhétoriques

Relativement peu de recherches ont été faites pour combiner la Théorie des actes de langage avec des cadres formels présents dans les théories en analyse et représentation du discours, cela parce que d'habitude les actes de langage sont considérés plutôt comme caractéristique d'un énoncé pris en isolation [156], [185], [164], tandis que les relations de discours connectent des paires, voire même des ensembles d'énoncés [11], [107], [90]. Cependant, si nous gardons les deux notions que nous tentons de mettre en valeur dans un cadre unifié, nous n'avons qu'à gagner, car les paires de types d'actes de langage peuvent constituer des contraintes supplémentaires sur l'ensemble des relations rhétoriques qui peuvent relier deux énoncés. Cela parce que dans des théories telles que la SDRT [9], [11] ou la RST [107] nous calculons les relations entre des énoncés seulement à partir des contenus (propositionnels) de ces énoncés, soit représentés sous forme sémantique (comme en SDRT), soit sous une forme hybride qui combine des aspects sémantiques avec des indices de surface (comme des connecteurs discursifs par exemple), comme dans l'approche de Marcu [108], basée sur la RST. Nous voyons donc comment les aspects illocutoires (exprimés au niveau du type de l'acte de langage réalisé par un énoncé) ne sont en général pas pris en compte de manière directe en structuration du discours. Cela même si par exemple la SDRT essaye d'intégrer l'illocutoire *via* les actes de langage indirects (de type Searle), mais l'appareillage formel mis en œuvre, quoiqu'assez complexe, ne nous semble pas offrir un traitement homogène de ce type de phénomènes (car, par exemple, les aspects illocutoires des énoncés ne concernent pas que les actes indirects).

A partir de ces constats, nous avons essayé de valider, en étudiant des corpus de dialogue homme-machine obtenus *via* la méthode du magicien d'Oz [194], [122], ou des dialogues réels entre des humains, les correspondances qui existent entre des paires de types d'actes de langage, au sens de la taxinomie de Searle [156], adaptée par Caelen au dialogue homme-machine [35]. Caelen et Xuereb ont été en mesure de proposer des tableaux d'association entre des paires d'actes et des relations rhétoriques [194] dans le cadre de la SDRT.

La taxinomie des *types* d'actes de langage proposée par Caelen suppose que le dialogue homme-machine est une coordination d'actions (linguistiques ou non), conformément à cer-

taines règles (afin d'atteindre un but présent ou futur) et supposant la co-construction de connaissances et *savoir-faires* mutuelles. Par conséquent, l'interaction se déroule *via* un échange d'actes (la notion d'actes de Caelen – que nous avons adoptée nous-mêmes – généralise la notion de Searle en y incluant aussi des actes non-verbaux – par exemple, des actions, mais dans cette thèse seuls les actes de langage seront considérés) ; chaque acte est en fait le résultat d'une application du domaine des contenus propositionnels au produit cartésien entre ces contenus propositionnels et un ensemble de *points illocutoires* (selon la notion de Vanderveken [185]). Les points illocutoires, que nous appellerons désormais **types d'actes de langage** sont régis par des axiomes et règles spécifiques, ayant des préconditions et des effets. Selon Caelen, certains actes sont *per-formés* (actionnels) afin de déterminer des changements dans l'état des choses – F^A (dénommé « faire-action ») pour réaliser une action (non-linguistique), F^F (dénommé « faire-faire ») pour déterminer (l'allocutaire) de faire une action ; d'autres actes sont épistémiques dans leur nature, c'est-à-dire, ils essayent de produire des changements dans les états mentaux des allocutaires – F^S (dénommé « faire-savoir ») pour informer l'allocutaire sur quelque chose, F^{FS} (dénommé « faire-faire-savoir ») pour demander des informations à l'allocutaire. Enfin, il y a deux types d'actes qui sont déontiques dans leur nature, c'est-à-dire, ils créent des obligations (des nécessités) ou donnent des choix (des possibilités) – F^D (dénommé « faire-devoir ») pour obliger l'allocutaire de faire quelque chose, et F^P (dénommé « faire-pouvoir ») pour offrir des choix à l'allocutaire. Du fait que cette taxinomie d'actes a été discutée *in extenso* dans d'autres articles [194], ouvrages [35] ou thèses [122], nous n'allons pas insister sur leur présentation dans cette thèse.

En revanche, pour les buts de notre recherche, nous insistons sur le fait que, dans notre cadre, chaque énoncé est caractérisé, d'un point de vue illocutoire, par un type d'acte de langage ; ainsi, chaque paire d'énoncés est caractérisée par une paire d'actes de langage et, d'un point de vue rhétorique, par un ensemble de relations rhétoriques reliant ces deux énoncés¹¹.

L'idée fondamentale concernant le lien entre les actes de langage et les relations rhétoriques est que l'ensemble des relations qui peuvent connecter une paire d'énoncés est contraint non seulement par les sémantiques des énoncés, mais aussi par les types d'actes de langage qui caractérisent ces énoncés d'un point de vue illocutoire (ou, autrement dit, que ces énoncés réalisent). Une étude sur corpus concernant la manière dont les actes agissent comme contraintes sur les relations rhétoriques a été réalisée par Xuereb et Caelen et présentée de manière détaillée en [194], donc ici nous ne donnerons que quelques exemples illustratifs.

Ainsi, supposons qu'il y a deux locuteurs, un sujet humain U et la machine M , et que U essaye de réserver un livre dans une bibliothèque. Donc, U demande au bibliothécaire (c'est-à-dire, à M) où il peut trouver un certain livre ; la machine peut donner plusieurs réponses, caractérisées par des actes de langage différents et reliées *via* des relations rhétoriques différentes à l'énoncé de U . L'exemple est indiqué ci-dessous (à gauche nous montrons l'acte de langage réalisé par l'énoncé, au milieu, l'énoncé même, et à droite, la relation rhétorique reliant cet

¹¹ Comme nous le verrons tout au long de cette thèse, nous essayerons de faire à ce que cet ensemble de relations rhétoriques reliant une paire d'énoncés soit réduit à une seule relation, cela pour des considérations à la fois pratiques – réduction de la complexité des structures rhétoriques manipulées, et théoriques – prise en compte des relations rhétoriques qui assurent un maximum de cohérence.

énoncé à l'énoncé provenant de U)¹² :

F_U^{FS} :	Où peux-je trouver 'L'alchimiste' de Coelho ?	
Réponses		
possibles de M		
F_M^S :	Ce livre se trouve au fond de ce couloir	<i>QAP</i>
	Juste à côté de la sortie, à gauche	<i>IQAP</i>
	Le plan des rangées de livres se trouve dans le hall d'entrée	<i>P-Elab</i>
F_M^{FS} :	Est-ce pour un rapport scientifique que vous devriez écrire ?	<i>Elab_q</i>
	Est-ce un livre scientifique que vous cherchez ?	<i>Background_q</i>
	Ce livre peut être trouvé à la bibliothèque de l'université	<i>P-Corr</i>
F_M^F :	Vous allez au fond du couloir	<i>QAP</i>
	Vous allez dans un autre bâtiment pour cela, le bâtiment 'B'	<i>P-Corr</i>
	Vous allez regarder sur la rangée qui se trouve juste à côté de la rangée de DVDs	<i>IQAP</i>
F_M^D :	Il faut valider votre carte d'abord, Monsieur	<i>P-Elab</i>
F_M^P :	Vous pouvez emprunter soit l'édition papier, soit la version DVD de ce livre	<i>P-Elab</i>

En utilisant des exemples tirés des corpus de dialogues obtenus *via* la méthode du magicien d'Oz, exemples du même type que ceux présentés dans l'exemple ci-dessus, Caelen et Xuereb ont montré que pour chaque paire d'actes de langage en dialogue, seulement quelques (d'habitude, deux ou trois) relations rhétoriques sont utilisées pour relier les énoncés en question [194].

Ces résultats sont mis à profit pour raffiner l'ensemble des relations rhétoriques candidats dans le processus de mise à jour de la SDRS en dialogue, selon l'algorithme détaillé ci-dessous, qui représente une version améliorée de l'algorithme présenté dans le paragraphe précédent, au sens où maintenant les actes de langage sont pris en compte pour réduire l'ensemble des relations rhétoriques candidates lors de la mise à jour de la SDRS. Cependant, à la différence de la présentation donnée dans la section antérieure, nous allons présenter ici les étapes de traitement de manière plutôt informelle ; les étapes supplémentaires par rapport à l'algorithme précédent sont soulignées.

pour chaque énoncé α à ajouter à la SDRS dialogique :

1. lire sa forme logique correspondante $K(\alpha)$, *via* une requête au contrôleur de dialogue ;
2. **pour** chaque énoncé β qui se trouve déjà dans la SDRS dialogique :
 - (a) lire sa forme logique $K(\beta)$;

¹²L'acte F^A n'est pas pris en compte dans cette thèse, puisqu'il n'est pas linguistique.

- (b) lire la paire $(\gamma_\alpha, \gamma_\beta)$ d'actes de langage pour cet énoncé et pour l'énoncé lu à l'étape 1 ;
- (c) déterminer l'ensemble P des relations rhétoriques *autorisées* par la paire d'actes lue à l'étape 2.(a) ;
- (d) **pour** chaque relation rhétorique ρ dans l'ensemble P :
 - i. lire la sémantique Σ_ρ de la relation rhétorique ρ ;
 - ii. calculer la valeur de vérité γ de la proposition $\Sigma_\rho(K(\alpha), K(\beta))$;
 - iii. **si** $\gamma = \text{FALSE}$, **alors aller** à l'étape 2.(c) ;
sinon, ajouter ρ à l'ensemble des relations rhétoriques dans la SDRS, et α à l'ensemble des énoncés dans cette SDRS, et **aller** à l'étape 2.(c).

L'algorithme présenté ci-dessus offre deux avantages par rapport à la procédure précédemment décrite : (i) la possibilité de filtrer *en amont* les relations rhétoriques calculées sur des bases sémantiques, cela pour réduire la complexité des calculs, et (ii) de manière connexe au point antérieur, l'obtention d'une certaine réduction du temps de calcul, en moyenne de 2-3 fois, car, au lieu de vérifier les sémantiques de 7 ou 10 relations rhétoriques (selon le fait que les énoncés ont été produits ou non par le même locuteur), nous en vérifions un sous-ensemble, en moyenne de 2 ou 3 relations seulement. Néanmoins, les réductions concernant le temps de calcul ne constituent qu'un avantage marginal de l'utilisation des actes de langage comme contraintes sur les relations rhétoriques « calculables » entre deux énoncés ; l'avantage essentiel de cette approche est qu'elle permet de filtrer des relations rhétoriques valables par le calcul sémantique, mais qui ne sont pas en accord avec les prédictions linguistiques. Cela parce que, les expériences sur des corpus de dialogues l'ont montré, les sémantiques des relations rhétoriques sous-spécifient au sens où elle sont parfois trop permissives : elles autorisent la présence d'une relation rhétorique entre deux énoncés, là où les interprétations intuitives des sémantiques des relations rhétoriques ne l'autoriseraient pas. Nous verrons un exemple dans ce sens dans le paragraphe suivant.

3.3.2 Calcul des relations rhétoriques : exemple étendu

Afin d'illustrer la capacité des actes de langage de filtrer les relations rhétoriques lors de la mise à jour des SDRS dialogiques, nous considérons un exemple de dialogue entre un système (qui joue le rôle d'un bibliothécaire virtuel – noté comme d'habitude par M) et un usager, client de la bibliothèque (noté par U) ; l'étiquette π_i désigne le i ème tour de parole¹³. Nous considérons un dialogue déjà existant entre les deux interlocuteurs :

U : Où est-ce que je peux trouver un bouquin sur l'astronomie ? ^{π_1}

M : Vous voulez un livre sur l'astronomie écrit par qui, Monsieur ? ^{π_2}

U : Qu'est-ce qu'il y a de dispo ? ^{π_3}

¹³Ici, chaque tour de parole contient un énoncé, donc nous pouvons considérer que la distinction tour de parole / énoncé, pour cet exemple particulier, n'est pertinente que dans la mesure où elle permet de distinguer les émetteurs des énoncés, et ainsi, de préciser l'ensemble des relations rhétoriques dont la capacité de connecter les énoncés est à vérifier.

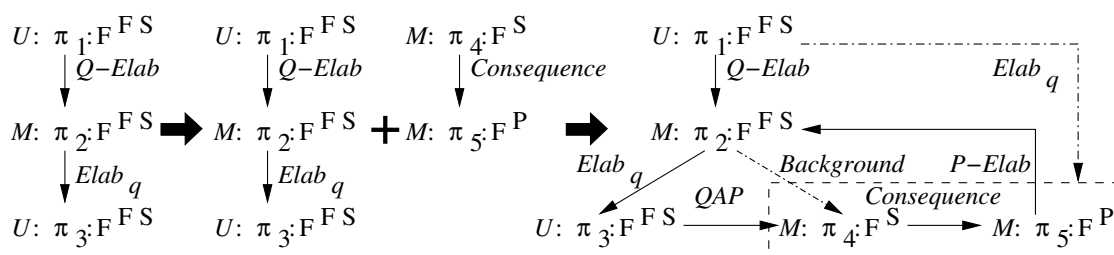


FIG. 3.1 – Mise à jour de la structure rhétorique

A partir de ce moment, la machine est censée répondre que les livres écrits par les auteurs Afin et Baffin sont disponibles sur l’astronomie et offrir à l’usager l’opportunité de choisir entre ces deux auteurs ; cela amène M à produire deux énoncés, le premier comme un acte d’informer l’usager (donc, un F^S) et le deuxième comme un acte de donner un choix à l’usager (donc, un F^P). Nous considérons que ces deux énoncés ne sont disponibles que sous forme logique (fournie par le contrôleur de dialogue) ; cependant, pour alléger la présentation, nous présentons deux formes linguistiques possibles pour ces énoncés, que nous montrons, en italiques, ci-dessous :

M : Nous avons des livres de Afin et Baffin sur ce sujet^{π4}.

M : Quel auteur préférez vous ?^{π5}

A partir des formes logiques de ces deux derniers énoncés, libellés respectivement π₄ et π₅, la machine construit une sous-SDRS qu’elle adjoindra à la SDRS du dialogue (c’est-à-dire, celle construite à partir des énoncés π₁, π₂ et π₃). Ce processus est illustré, par étapes, dans la figure 3.1, où nous avons noté, séparés par « : », le locuteur, l’étiquette de l’énoncé et le type d’acte de langage réalisé par l’énoncé ; les relations rhétoriques entre les énoncés sont marquées par des flèches, leurs noms étant donnés à droite des flèches. Les lignes en pointillée marquent les relations rhétoriques déterminées comme valides par le calcul sémantique, mais non-autorisées par le filtre présenté ici (c’est-à-dire, par la paire d’actes réalisés par ces énoncés). Ainsi, lorsque la machine relie π₄ à π₅ via une relation rhétorique, seule la relation de *Consequence* est autorisée par la paire d’actes F^S et F^P dans un contexte monologique. Ensuite, la SDRS ainsi formée avec ces deux énoncés et la relation de Conséquence entre eux, est reliée à la SDRS du dialogue via plusieurs relations rhétoriques, calculées sur des bases sémantiques :

- *QAP* est trouvée entre les énoncés π₃ et π₄ ;
- *Background* est trouvée entre les énoncés π₄ et π₂ ;
- *P-Elab* est trouvée entre les énoncés π₂ et π₅ ;
- *Elab_q* est trouvée entre l’énoncé π₁ et le constituant discursif formé par la paire d’énoncés (π₄, π₅).

De ces relations, *Background*(π₄, π₂) et *Elab_q*(π₁, (π₄, π₅)) ne sont pas autorisées par les paires d’actes de langage réalisés par les énoncés en question, ce qui correspond à la fois à nos intuitions et aux sémantiques informelles de ces relations rhétoriques en SDRT [11].

Pour la complétude des traitements, nous présentons ci-dessous de manière détaillée le calcul(sémantique) des relations *Background*(π₄, π₂) et *P-Elab*(π₂, π₅). Pour ce faire, nous précisons aussi (ci-dessous) les formes logiques de ces énoncés :

- $\pi_2 \mapsto K(\pi_2) ::= \exists X, Y, T, V : \text{agent}(X) \wedge \text{equals}(X, U) \wedge \text{equals}(X, \neg\text{emitter}(\pi_2)) \wedge \text{object}(Y) \wedge \text{equals}(Y, \text{'book'}) \wedge \text{feature}(Y, T) \wedge \text{author}(T) \wedge \text{equals}(T, \text{'?'}) \wedge \text{feature}(Y, V) \wedge \text{field}(V) \wedge \text{equals}(V, \text{'astronomy'}) \wedge \text{want}(X, Y) \wedge \text{equals}(\Delta t(\pi_2), t)$;
- $\pi_4 \mapsto K(\pi_4) ::= \exists X, Y, Z, T, V, W : \text{agent}(X) \wedge \text{equals}(X, M) \wedge \text{equals}(X, \text{emitter}(\pi_4)) \wedge \text{object}(Y) \wedge \text{equals}(Y, \text{'book'}) \wedge \text{object}(Z) \wedge \text{equals}(Z, \text{'book'}) \wedge \text{feature}(Y, T) \wedge \text{author}(T) \wedge \text{equals}(T, \text{'afin'}) \wedge \text{feature}(Z, V) \wedge \text{author}(V) \wedge \text{equals}(V, \text{'baffin'}) \wedge \text{ExhaustiveDecomposition}(Y, Z, W) \wedge \text{have}(X, W) \wedge \text{equals}(\Delta t(\pi_4), t_V)$;
- $\pi_5 \mapsto K(\pi_5) ::= \exists X, Y, Z : \text{agent}(X) \wedge \text{equals}(X, U) \wedge \text{equals}(X, \neg\text{emitter}(\pi_5)) \wedge \text{object}(Y) \wedge \text{equals}(Y, \text{book}) \wedge \text{feature}(Y, Z) \wedge \text{author}(Z) \wedge \text{equals}(Z, \text{'?'}) \wedge \text{want}(X, Y) \wedge \text{equals}(\Delta t(\pi_5), t)$.

Afin de prouver, par exemple, que la relation $\text{Background}(\pi_4, \pi_2)$ est vraie, il faut parcourir les étapes suivantes :

1. substitutions des variables, ainsi que $K(\pi_4)$ et $K(\pi_2)$ d'une part, et $K(\pi_2)$ et $K(\pi_5)$ d'autre part, n'ont pas de variables avec les mêmes noms ; donc, en $K(\pi_2)$, les substitutions suivantes sont réalisées : $X'/X, Y'/Y, Z'/Z, T'/T, V'/V$;
2. représentation explicite de la sémantique de la relation rhétorique couramment vérifiée : $\text{Background}(\pi_4, \pi_2) ::= \text{enounce}(\pi_4) \wedge \text{entails}(\pi_4, \pi_2) \mapsto \text{enounce}(\pi_4) \wedge (K(\pi_4) \Rightarrow K(\pi_2)) \wedge \text{equals}(\Delta t(\pi_4), t_V)$;
3. vérification de chaque proposition « atomique » dans la sémantique de la relation rhétorique (c'est-à-dire, de chaque proposition séparée par le connecteur \wedge) : pour $\text{Background}(\pi_4, \pi_2)$:
 - $\text{enounce}(\pi_4)$: cette proposition est vraie, car π_4 ne contient pas de variables non-initialisées (c'est-à-dire, dont la valeur soit égale à '?') ;
 - $\text{equals}(\Delta t(\pi_4), t_V)$: cette proposition est aussi vraie, de la sémantique de l'énoncé π_4 (c'est-à-dire, des propositions « atomiques » en $K(\pi_4)$) ;
 - $K(\pi_4) \Rightarrow K(\pi_2)$: afin de prouver cette proposition, il faut parcourir les étapes suivantes :
 - (a) élimination des quantificateurs existentiels dans $K(\pi_4)$ et $K(\pi_2)$:
 - $K(\pi_4) ::= \text{equals}(M, \text{emitter}(\pi_4)) \wedge (\text{author}(\text{'afin'}) \wedge \text{feature}(\text{book}, \text{'afin'}) \wedge \text{have}(M, \text{book})) \wedge (\text{author}(\text{'baffin'}) \wedge \text{feature}(\text{book}, \text{'baffin'}) \wedge \text{have}(M, \text{book})) \wedge \text{equals}(\Delta t(\pi_4), t_V)$;
 - $K(\pi_2) ::= \text{equals}(U, \neg\text{emitter}(\pi_2)) \wedge \text{author}(\text{'?'}) \wedge \text{field}(\text{'astronomy'}) \wedge \text{feature}(\text{book}, \text{'?'}) \wedge \text{feature}(\text{book}, \text{'astronomy'}) \wedge \text{want}(U, \text{book}) \wedge \text{equals}(\Delta t(\pi_2), t)$;
 - (b) preuve de $K(\pi_4) \Rightarrow K(\pi_2)$: $K(\pi_4) \Rightarrow K(\pi_2) \Leftrightarrow \neg K(\pi_4) \vee K(\pi_2)$, ce qui est vrai si et seulement si la proposition $K(\pi_4) \wedge \neg K(\pi_2)$ est fautive ; la fausseté de la dernière proposition logique résulte de sa confrontation avec l'ontologie de tâches.

La relation rhétorique $P - \text{Elab}(\pi_2, \pi_5)$ est calculée de manière similaire ; cependant, les valeurs de vérité pour ces énoncés sont calculées de gauche à droite, donc dans la sémantique de $P - \text{Elab}$ il suffit de ne vérifier qu'une partie :

$P - \text{Elab}(\pi_2, \pi_5) ::= \text{good_time}(\Delta t(\pi_5)) \wedge \neg \text{Disjoint}(\Delta t(\pi_5), \text{SARG}(\pi_2)) \wedge \text{ClassOf}(K(\pi_5), \text{Plan}) \wedge (\text{Plan} \Rightarrow \text{SARG}(\pi_2))$,

où l'objet Plan réunit des parties dans la sémantique de l'énoncé π_5 ainsi qu'il implique le SARG de π_2 , $\text{want}(U, \text{book}) \wedge \text{feature}(\text{book}, \text{'F'}) \wedge \text{feature}(\text{book}, \text{'?'}) \wedge \text{field}(\text{'astronomy'}) \wedge \text{author}(\text{'?'})$.

3.4 Conclusions

Dans ce chapitre nous avons présenté certains aspects contextuels qui peuvent être pris en compte à un niveau sémantique, afin de pouvoir structurer, du point de vue rhétorique, les répliques dialogiques. Ce contexte est formalisé, d'un côté, au niveau d'une ontologie de tâches qui code le domaine de l'application¹⁴ et, d'un autre côté, au niveau discursif, par l'ensemble des énoncés et des relations rhétoriques qui existent entre ces énoncés, à un moment donné. Ainsi, nous avons vu comment, en spécifiant des sémantiques formelles pour les relations rhétoriques, nous arrivons à calculer une structure du dialogue, cela en regardant le problème comme une mise à jour d'une structure existante (éventuellement vide), plutôt que comme une construction à partir de zéro d'une telle structure, pour un discours donné. Autrement dit, dans notre approche, le contenu du discours devient un modèle, que nous mettons ensuite à jour, en le *complétant*. Donc, nous n'enlevons pas les relations rhétoriques calculées auparavant, parce que nous gardons la trace de toutes les contributions dialogiques, y compris en situations de contradiction.

Pour ce qui tient aux remises en question ou aux remises en cause en dialogue, le caractère révisable des relations rhétoriques inférées n'est pas modélisé de manière explicite. La structure rhétorique garde toutes les relations calculées car, nous insistons sur ce fait, notre but n'est pas d'obtenir une interprétation aussi cohérente que possible d'un discours donné d'avance, mais de produire une trace aussi fidèle que possible de la succession des tours de parole en dialogue.

Par la suite, nous avons présenté une manière dont le lien entre les actes de langage réalisés par les énoncés et les relations rhétoriques qui relient des paires d'énoncés, peut être exploité pour contraindre davantage l'ensemble des relations qui peuvent connecter les énoncés. Cela aide notamment à éliminer certaines relations qui sont autorisées par le calcul sémantique, mais en désaccord avec les prédictions linguistiques. En fait, nous essayons de contourner certaines des limites dans notre algorithme de mise à jour de la SDRS dialogique, en y ajoutant des contraintes au niveau des actes de langage réalisés par les énoncés. Ces actes sont calculés au préalable soit par l'interpréteur pragmatique (voir chapitre 1) pour les énoncés provenant des usagers, soit par le contrôleur de dialogue pour les énoncés censés être générés par la machine.

Dans le chapitre suivant nous verrons comment on peut compléter les contraintes sur les relations rhétoriques calculées, en renforçant le rôle du contexte dans le *choix* de ces relations.

¹⁴La conception des ontologies de tâche ne fait pas l'objet de cette thèse, mais nous renvoyons le lecteur aux travaux de Caelen [35] ou de Nirenburg [124] pour des discussions approfondies sur ce point.

Chapitre 4

Filtrage contextuel des relations rhétoriques en dialogue

4.1 Introduction

Dans le chapitre 3 nous avons décrit de manière détaillée les étapes de filtrage local (sémantique et pragmatique) des relations rhétoriques ; dans ce chapitre nous allons décrire *in extenso* une étape supplémentaire de filtrage contextuel des relations rhétoriques.

Ce filtrage contextuel consiste à parcourir trois étapes : (a) chaque relation rhétorique est exprimée en termes de prédicats discursifs appliqués sur les étiquettes des énoncés (comme dans le chapitre 3) ; (b) chaque formule sémantique ainsi obtenue est convertie dans un programme logique (LP), représenté selon les conventions du langage Prolog ; (c) chaque LP ainsi obtenu est converti dans une grammaire abstraite d'attributs (AAG – « Abstract Attribute Grammar ») [53], [83].

Pour les situations où plus d'une relation peut encore relier une paire d'énoncés après le filtrage contextuel, nous proposons un mécanisme (optionnel) de filtrage numérique des relations, ce qui peut nous permettre de garder, pour chaque mise à jour d'une structure rhétorique, seulement une relation rhétorique entre deux énoncés. Le fait de ne garder qu'une relation rhétorique entre deux énoncés est utile du point de vue computationnel, en nous permettant de réduire la complexité de la recherche dans les graphes SDRS des dialogues.

Ainsi, après un bref aperçu sur la relation entre les formes sémantiques des relations rhétoriques et les grammaires abstraites d'attributs, nous allons présenter de manière détaillée le mécanisme de filtrage contextuel, en présentant aussi les fondements théoriques de ce cadre, ainsi qu'une discussion sur ses limites et des extensions potentielles. Enfin, la méthode proposée sera illustrée *via* un exemple détaillé, tiré d'un dialogue réel, un fragment d'une scène dans une pièce de théâtre.

4.2 Des relations rhétoriques aux grammaires abstraites d'attributs

4.2.1 Relations de discours et programmes logiques

La conversion des sémantiques des relations rhétoriques en programmes logiques qui ont les étiquettes des énoncés comme des variables en entrée est assez immédiate, en rendant graduellement explicites les prédicats utilisés (voir chapitre 3). Ainsi, pour un traitement général, nous supposons une relation rhétorique libellée ρ et exprimée en termes des prédicats p_1 à p_n (α et β désignent les énoncés) :

$$\rho(\alpha, \beta) ::= p_1 \xi_1 p_2 \xi_2 \dots \xi_{n-1} p_n,$$

où ξ_i pour i de 1 à $n - 1$ représentent les connecteurs logiques entre les prédicats, ainsi que les quantificateurs sur les variables présentes ; ces connecteurs sont \wedge , \vee , \neg et \Rightarrow ; les quantificateurs sont \exists et \forall . Chacun des prédicats p_i est à son tour exprimé en termes d'autres foncteurs t_j^i , avec j de 1 à n_i . Nous supposons aussi que les arguments des prédicats p_i sont des termes de la forme s_k^i et qu'une partie de ces termes prennent α et β comme arguments. En numérotant les indices inférieurs pour les termes s , la formule présentée ci-dessus peut être écrite comme :

$$\rho(\alpha, \beta) ::= p_1(s_1^1(\alpha, \dots), s_2^1(\beta, \dots), \dots) \xi_1 \dots \xi_{n-1} p_n(s_1^n(\alpha, \dots), s_2^n(\beta, \dots), \dots).$$

De manière similaire, en supposant que parmi les termes t il y a quelques-uns qui prennent α et β comme arguments, et en numérotant ces termes, nous obtenons une nouvelle définition pour chacun des prédicats p_i , qui est ensuite étendue, de façon récursive, jusqu'à ce que les prédicats discursifs sont utilisés (ici, les ξ_j^i représentent une numérotation des ξ_j , donc ils sont aussi des connecteurs ou des quantificateurs) :

$$p_i(s_1^i(\alpha, \dots), s_2^i(\beta, \dots), \dots) ::= t_1^i(\alpha \dots) \xi_1^i t_2^i(\beta, \dots) \xi_2^i \dots \xi_{n_i-1}^i t_{n_i-1}^i(\dots).$$

Afin de transformer ces spécifications dans des programmes logiques (à la Prolog), nous devons remplacer $::=$ par $:$, et réaliser une correspondance μ de l'ensemble des connecteurs et quantificateurs, dans l'ensemble des balises légales en Prolog :

$$\mu : \{\wedge \vee \neg \Rightarrow \exists \forall\} \rightarrow \{, ; \backslash +\}.$$

Pour les trois premiers éléments dans cet ensemble, la correspondance est immédiate ($\mu(\wedge) = ,,$, $\mu(\vee) = ;$ et $\mu(\neg) = \backslash +$), tandis que pour les derniers trois, en notant par \uparrow l'opérateur de concaténation, nous avons que $\mu(\Rightarrow) = \mu(\neg) \uparrow \mu(\vee) = \backslash + \uparrow$ ¹. En ce qui concerne les quantificateurs, compte tenu du comportement de Prolog [63], \exists peut être éliminé et \forall remplacé par le prédicat `setof/3`².

Avec ce formalisme nous présentons ci-dessous un exemple concret de programme logique obtenu à partir de la sémantique de la relation QAP , en supposant que les énoncés représentent les variables d'entrée de ce programme. Nous partons de la sémantique de QAP , que nous reprenons ici du chapitre 3, pour faciliter la lecture :

$$QAP(\alpha, \beta) ::= \text{question}(\alpha) \wedge \text{enounce}(\beta) \wedge \neg \text{Disjoint}(\text{topic}(\alpha), \text{topic}(\beta)) \wedge \text{answer}(\beta, \alpha).$$

Ensuite, le programme est détaillé ci-dessous, en éclatant chaque prédicat discursif en prédicats Prolog :

¹Cela est basé sur le fait que, du point de vue logique, la clause $\alpha \Rightarrow \beta$ est équivalente à $\neg \alpha \vee \beta$.

²Plus précisément, une clause de la forme $\forall X : p(X)$ peut être remplacée par `setof(X, p(X), Set)`, où la variable `Set` rassemble toutes les instanciations de X telles que $p(X)$.

- (1) $QAP(\alpha, \beta) : -\text{question}(\alpha), \text{enounce}(\beta), \setminus + \text{Disjoint}(\text{topic}(\alpha), \text{topic}(\beta)), \text{answer}(\beta, \alpha).$
- (2) $\text{enounce}(\alpha) : -\text{memberOf}(X, K(\alpha)), \text{memberOf}(Y, \Omega), X == Y.$
- (3) $\text{question}(\alpha) : -\setminus + \text{enounce}(\alpha).$
- (4) $\text{memberOf}(X, Y) : -\text{member}(X, Y).$
- (5) $K(X) : -X = ..L, L.$
- (6) $\text{Disjoint}(X, Y) : -\text{memberOf}(_X, X), \text{memberOf}(_Y, Y), _X = \setminus = _Y.$
- (7) $\text{topic}(X) : -\text{ExhaustiveDecomposition}(\phi(_i), \omega(_j)), \text{memberOf}(\phi(_i), K(X)),$
 $\text{memberOf}(\omega(_j), \Omega), \phi(_i) == \omega(_j).$
- (8) $\text{ExhaustiveDecomposition}(X, Y) : -\text{memberOf}(_X, X), \text{memberOf}(_Y, Y),$
 $\text{memberOf}(_X, \Omega),$
 $\text{memberOf}(_Y, \Omega), \text{memberOf}(_ \omega, \Omega),$
 $(\text{memberOf}(_ \omega, X);$
 $\text{memberOf}(_ \omega, Y)), (\text{memberOf}(_X, X) - >$
 $\setminus + \text{memberOf}(_X, Y); \text{memberOf}(_Y, Y) - >$
 $\setminus + \text{memberOf}(_Y, X)).$
- (9) $\text{answer}(X, Y) : -\text{topic}(X) == \text{topic}(Y), t(X) > t(Y), \text{setof}(_ \nu, \text{member}(_ \nu, K(\beta)), \text{Set}),$
 $\text{member}(_ \omega, \Omega), \text{member}(_ \nu, \text{Set}), _ \nu == _ \omega.$
- (10) $SARG(X) : -\text{setof}(_X, K(X), \text{SetX}); \text{SetX} == K(X), \text{memberOf}(\gamma, \Gamma),$
 $\setminus + \text{Disjoint}(\gamma, \text{SetX}).$
- (11) $\text{emitter}(X) : -\setminus + \text{memberOf}(u, K(X)) - > X = u; \setminus + \text{memberOf}(m, K(X)) - > X = m.$

Les ensembles Ω et Γ représentent, respectivement, l'ensemble des concepts dans l'ontologie de tâches [35] (mais voir aussi chapitre 1), et l'ensemble des types d'actes de langage (voir chapitre 3). Ces ensembles peuvent être assertés comme des entités Prolog :

$:- \text{assert}(\Omega = [/*Concepts list*/]). \quad :- \text{assert}(\Gamma = [F^F, F^S, F^{FS}, F^P, F^D]).$

4.2.2 Aperçu grammatical des relations rhétoriques

La correspondance entre les programmes logiques et les grammaires d'attributs est un sujet bien étudié depuis plus d'une décennie [53]. La différence entre les grammaires d'attributs et les grammaires abstraites d'attributs réside dans les attributions des variables qui accompagnent chaque règle de production : tandis que dans les grammaires d'attributs il y a une attribution par *égalité* pour chaque variable, dans les grammaires abstraites d'attributs les variables sont spécifiées *via* des *ensembles de restrictions* qui contiennent toutes les variables qui s'unifient dans la même valeur (connue ou non). Cette différence est importante, car les grammaires d'attributs ont besoin d'une séparation préalable des variables en *héritées* (c'est-à-dire, d'entrée) et *synthétisées* (c'est-à-dire, de sortie), tandis que les grammaires abstraites d'attributs (d'où leur nom) n'ont pas besoin de cette séparation. Cela est utile notamment pour une représentation grammaticale des programmes logiques où il n'y a pas encore de séparation entre les variables d'entrée et de sortie (donc, pour lesquelles aucune requête n'a encore été formulée). Cela peut être assimilé à la situation où un programme logique qui correspond à une relation rhétorique est manipulé pour toute paire possible d'énoncés (c'est-à-dire, les énoncés sont gardés comme des variables, considérées non-instanciées).

Dans le cas général, une transformation d'un programme logique (qui correspond à la sémantique d'une relation rhétorique) dans une grammaire abstraite d'attributs suppose la construction

d'une règle de production et d'une série d'ensembles de restrictions (un tel ensemble de restrictions pour chaque variable dans la règle de production d'origine). Ainsi, pour une règle de la forme :

$$\rho(_ \alpha, _ \beta) : -p_1(s_1^1(_ \alpha, \dots), s_2^1(_ \beta, \dots), \dots)\mu(\xi_1)\dots\mu(\xi_{n-1})p_n(s_1^n(_ \alpha, \dots), s_2^n(_ \beta, \dots), \dots),$$

nous obtenons la règle de production suivante, dans une AAG :

$$\rho \rightarrow p_1\tau(\mu(\xi_1))\dots\tau(\mu(\xi_{n-1}))p_n,$$

où τ est une fonction réalisant la correspondance entre les connecteurs Prolog et une notation BNF (« Backus-Naur Form ») : $\tau : \{, ; \backslash +\} \rightarrow \{, \}$, et $\tau(\backslash + p) = q$, ainsi que $q \equiv \neg p$.

Afin de construire les ensembles de restrictions pour les variables (c'est-à-dire, pour les attributs) qui apparaissent dans la règle Prolog d'origine, nous définissons les fonctions *sélecteur* λ_m^f : pour un foncteur f d'arité M , λ_m^f retourne le m -ème argument de f , si $m \leq M$, et \emptyset sinon. Ainsi, par exemple, si nous avons $f(t_1, \dots, t_m, \dots, t_M)$, alors $\lambda_m^f = t_m$. Ces fonctions sélecteur peuvent être composées : si, à son tour, t_m est d'arité N , nous pouvons sélectionner son n -ème argument avec $\lambda_n^{t_m}$, pour $n \leq N$: si nous avons $t_m(s_1, \dots, s_n, \dots, s_N)$, alors $\lambda_m^f \circ \lambda_n^{t_m} = s_n$. Ici, la relation de composition \circ est définie par $f \circ g(x) = g(f(x))$ [53].

Ainsi, à l'instar de Isakowitz [83], nous notons les éléments dans les ensembles de restrictions pour une règle par $p.i(j)$, où p est le prédicat où la variable apparaît, i est l'indice (la position) de la variable dans le prédicat p , et j désigne la *profondeur* du prédicat : j est égal à ϵ si le prédicat p est dans la partie gauche de la règle de production, 1 si le prédicat est juste à droite du symbole \rightarrow dans la règle de production, et t s'il y a $t - 1$ termes (séparés par des connecteurs) entre le symbole \rightarrow et le prédicat p . De plus, nous notons par *id* la fonction d'identité : $id(X) = X$, quel que soit le type de X (terme ou atome). Par conséquent, pour la règle Prolog générale considérée ci-dessus, nous avons les ensembles de restrictions suivants pour les variables $_ \alpha$ et $_ \beta$:

$$_ \alpha : \{\rho.1(\epsilon), \lambda_1^{s_1^1} \circ id(p_1.1(1)), \dots, \lambda_1^{s_1^n} \circ id(p_n.1(n))\};$$

$$_ \beta : \{\rho.2(\epsilon), \lambda_1^{s_1^1} \circ id(p_1.2(1)), \dots, \lambda_1^{s_1^n} \circ id(p_n.2(n))\}.$$

Nous allons illustrer l'approche sur un exemple, concernant toujours la relation rhétorique *QAP*. Pour chaque ligne dans le programme logique il y a une ligne contenant une règle de production, et une « ligne » contenant les ensembles de restrictions pour les variables concernées. Tout d'abord, nous présentons les règles de production :

(1') *QAP* \rightarrow enounce, enounce, neg, confirmation

(2') enounce \rightarrow memberOf, memberOf, ==

(3') question \rightarrow neg

(4') memberOf \rightarrow member

(5') *K* \rightarrow = ..

(6') Disjoint \rightarrow memberOf, memberOf, neg

(7') topic \rightarrow ExDec, memberOf, memberOf, ==

(8') ExDec \rightarrow memberOf, memberOf, memberOf, memberOf, memberOf, memberOf,
neg | neg | neg | neg

(9') answer \rightarrow ==, >, setof, member, member, ==

(10') *SARG* \rightarrow setof, ==, memberOf, neg

(11') emitter \rightarrow Disjoint | = | Disjoint | =.

Le non-terminal `neg` désigne l'application de la fonction τ sur une construction du type $\neg p$, pour obtenir un q tel que $q \equiv \neg p$. Dans ces règles de production, nous considérons que les prédicats Prolog sont disponibles comme des terminaux (par exemple, dans les règles montrées ci-dessus, les prédicats `= ./2`, `==/2`, `=/2` et `member/2` sont des terminaux).

En ce qui concerne les ensembles de restrictions qui résultent pour les variables présentes dans les règles Prolog d'origine, nous listons ci-dessous ceux associés aux deux premières règles de production :

$$\begin{aligned} (1'') \quad _ \alpha &: \{id(QAP.1(\epsilon)), id(question(1)), \lambda^{neg} \circ \lambda_1^{Disjoint} \circ id(topic(3)), id(answer.2(4))\}; \\ _ \beta &: \{id(QAP.2(\epsilon)), id(enounce(2)), \lambda^{neg} \circ \lambda_2^{Disjoint} \circ id(topic(3)), id(answer.1(4))\} \\ (2'') \quad _ \alpha &: \{id(enounce(\epsilon)), \lambda_2^{memberOf} \circ id(K(1)), id(memberOf.1(1)), id(==.1(2))\}; \\ _ X &: \{id(memberOf.1(1)), id(==.1(2))\}; _ Y : \{id(memberOf.2(1)), id(==.2(2))\}. \end{aligned}$$

4.3 Filtrage contextuel des relations rhétoriques

4.3.1 Procédure de filtrage « syntaxique »

Les sémantiques des relations rhétoriques peuvent être utilisées de plusieurs manières en structuration rhétorique ; une de ces manières, le filtrage sémantique local (c'est-à-dire, à base de paires d'énoncés) des relations rhétoriques autorisées par les combinaisons d'actes de langage, a déjà été décrite dans le chapitre 3. Une autre manière de s'appuyer sur ces sémantiques peut consister à essayer d'évaluer la pertinence d'une relation rhétorique par rapport au discours entier, et non pas seulement à la paire d'énoncés reliés par cette relation. Une façon de faire cela, qui constitue en fait le noyau de ce chapitre, consiste à voir une structure rhétorique comme *cohérente* si et seulement si elle peut être *calculée*, via les sémantiques des relations rhétoriques et des énoncés. Cela nous amène à la définition suivante :

Définition 1. Une structure de discours (SDRS) est *cohérente* si elle peut être calculée, via un programme logique, en utilisant les sémantiques des relations rhétoriques.

D'un point de vue syntaxique, une SDRS, donnée comme point de départ dans un processus de mise à jour, contient :

- des énoncés ou des constituants discursifs : π_1, \dots, π_N ;
- des relations rhétoriques : ρ_1, \dots, ρ_M ;
- une structure : $SDRS_{MN} = \bigcup_{i=1}^M \bigcup_{j,k=1}^N \rho_i(\pi_j, \pi_k)$; de plus, il peut y avoir des $k \in \{1, \dots, N\}$ tels qu'il existe $l \in \{1, \dots, M\}$ et $m, n \in \{1, \dots, N\}$ ainsi que $\pi_k = \rho_l(\pi_m, \pi_n)$.

Si un énoncé, libellé π_{N+1} doit être ajouté à $SDRS_{MN}$, nous supposons que les processus cascades de filtrage pragmatique et sémantique ont fourni en sortie $R \geq 1$ relations rhétoriques telles que :

$$\forall i \in \{1, \dots, R\} \quad \forall r_i \in \{1, \dots, N\} \wedge |\langle r_i \rangle| = R : \rho_i(\pi_{r_i}, \pi_{N+1}) = \text{TRUE},$$

où $\langle x \rangle$ signifie l'ensemble des x possibles, satisfaisant certaines conditions spécifiées au préalable.

Dans ce cadre, le filtrage contextuel des relations rhétoriques consiste à parcourir les étapes suivantes :

1. Conversion de la structure $SDRS_{MN}$ dans un programme logique, en transformant chaque relation rhétorique dans le programme logique correspondant, pour ensuite *concaténer* ces programmes logiques :

$$SDRS_{MN} = \bigcup_{i=1}^M \bigcup_{j,k=1}^N \rho_i(\pi_j, \pi_k) \mapsto \uparrow_{i=1}^M \uparrow_{j,k=1}^N LP_{\rho_i}(\pi_j, \pi_k) = LP_{MN}.$$

La correspondance $\rho \mapsto LP_{\rho}$ est réalisée selon les discussions dans les paragraphes antérieurs : si

$\rho_i(\pi_j, \pi_k) ::= p_1(s_1^1(\pi_j, \dots), s_2^1(\pi_k, \dots), \dots) \xi_1 \dots \xi_{I-1} p_I(s_1^I(\pi_j, \dots), s_2^I(\pi_k, \dots), \dots)$, alors la clause correspondante dans le LP est :

$$\rho_i(\pi_j, \pi_k) : -p_1(s_1^1(\pi_j, \dots), s_2^1(\pi_k, \dots), \dots) \mu(\xi_1) \dots \mu(\xi_{I-1}) p_I(s_1^I(\pi_j, \dots), s_2^I(\pi_k, \dots), \dots) ;$$

2. Conversion du programme logique LP_{MN} correspondant à la SDRS, dans une grammaire abstraite d'attributs AAG :

$$LP_{MN} = \uparrow_{i=1}^M \uparrow_{j,k=1}^N LP_{\rho_i}(\pi_j, \pi_k) \mapsto \uparrow_{i=1}^M \uparrow_{j,k=1}^N AAG_{\rho_i}(\pi_j, \pi_k) = AAG_{MN}.$$

La transformation $LP_{\rho} \mapsto AAG_{\rho}$ est réalisée comme indiqué dans le paragraphe antérieur ; pour un LP comme celui montré à l'étape 1, nous avons des règles de la forme :

$$(1) \rho_i \rightarrow p_1 \tau(\mu(\xi_1)) \dots \tau(\mu(\xi_I)) p_I,$$

...

$$(K) p_I \rightarrow t_1^I \tau(\mu(\xi_1^I)) \dots$$

Les ensembles de restrictions correspondant aux variables (*globales* – voir ci-dessous) π_j et π_k sont de la forme :

$$(1') \pi_j : \{\rho_i.1(\epsilon), \lambda_1^{p_1} \circ id(s_1^1.1(1)), \dots, \lambda_1^{p_I} \circ id(s_1^I.1(I))\} ; \pi_k : \{\rho_i.2(\epsilon), \lambda_2^{p_1} \circ id(s_2^1.1(1)), \dots, \lambda_2^{p_I} \circ id(s_2^I.1(I))\} ;$$

...

$$(K') \pi_j : \{\lambda_1^{p_1} \circ id(s_1^1.1(\epsilon)), \lambda_{\dots}^{t_1^1} \circ \dots, \dots\} ; \pi_k : \{\lambda_2^{p_1} \circ id(s_2^1.1(\epsilon)), \lambda_{\dots}^{t_1^1} \circ \dots, \dots\}.$$

Ici, par « variable globale » nous entendons une variable logique qui ne change ni de signification, ni de valeur, d'une règle dans l'AAG, à l'autre.

3. Ajout de $AAG_{\rho_i}(\pi_{r_i}, \pi_{N+1})$, avec $i = 1, \dots, R$, à $AAG_{MN} : AAG_{M+1, N+1}(i) = AAG_{MN} \uparrow AAG_{\rho_i}(\pi_{r_i}, \pi_{N+1})$. Nous obtenons ainsi R grammaires abstraites d'attributs qui correspondent à R structures rhétoriques *possibles*, chacune avec les mêmes $N + 1$ énoncés, mais avec des ensembles différents de $M + 1$ relations rhétoriques³.
4. Vérification de l'unifiabilité, pour chacune des R AAG discursives : c'est-à-dire, pour chaque variable globale (énoncé) dans chaque grammaire, il faut *unifier* tous les ensembles de restrictions pour cette variable, afin que le programme logique ayant généré la grammaire soit consistant. Autrement dit, pour que le LP à partir duquel la AAG a été générée soit consistant (c'est-à-dire, il ne contient pas de règles mutuellement incompatibles – règles qui ne peuvent être simultanément vraies sous les mêmes attributions de variables), il faut s'assurer que tous les éléments dans tous les ensembles de restrictions pour chaque variable globale dans chaque grammaire $AAG_{M+1, N+1}(i)$ peuvent être unifiés. Par exemple, pour la variable globale π_j de l'étape 2, nous devrions avoir, si la relation rhétorique courante i est telle que $r_i = j$ (cela n'est pas garanti d'être vrai, mais s'il ne l'est pas, alors

³Ces ensembles de relations rhétoriques sont différents par un seul élément : la dernière relation rhétorique ajoutée.

nous choisissons un j pour lequel cela est vrai), un élément du type $\rho_o.1(\epsilon)$ (où ρ_o désigne une relation rhétorique plus « ancienne », c'est-à-dire, déjà présente dans $S\text{DRS}_{MN}$) peut être unifié avec un élément du type $\lambda_k^{p_j} \circ id(s_k^J.m(J))$. Donc, cela signifie que l'équation $\rho_o.1(\epsilon) = \lambda_k^{p_j} \circ id(s_k^J.m(J))$ devrait avoir des solutions. Nous allons voir plus loin dans ce chapitre, *via* un exemple étendu, que parfois cela est vrai, parfois cela ne l'est pas.

5. S'il existe une variable globale (en fait, un énoncé) dans $AAG_{M+1,N+1}(i)$ pour laquelle aucun des ensembles de restrictions ne peut être unifié (c'est-à-dire, il y a au moins deux éléments dans deux ensembles de restrictions différents qui ne peuvent pas être unifiés), cela signifie que $LP_{MN} \uparrow LP_{\rho_i(\pi_{r_i}, \pi_{N+1})}$ n'est pas consistant, ce qui revient au fait que la structure de discours $S\text{DRS}_{M+1,N+1}(i) = S\text{DRS}_{MN} \cup \{\pi_{N+1}, \rho_i(\pi_{r_i}, \pi_{N+1})\}$ n'est pas cohérente, en vertu du Lemme 1 (voir ci-dessous). Dans cette situation, la relation rhétorique $\rho_i(\pi_{r_i}, \pi_{N+1})$ est supprimée.

Un point très important à mettre en évidence est qu'à la différence de ce que nous avons présenté dans le paragraphe 4.2, dans le filtrage contextuel des relations rhétoriques nous n'avons pas besoin de considérer la grammaire abstraite d'attributs associée à la dérivation *complète* de la relation rhétorique, en étendant chaque prédicat discursif. Cela est le cas, puisque dans le filtrage sémantique local chaque relation rhétorique avait déjà été validée, donc chaque prédicat discursif a été déterminé comme étant vrai. En filtrage contextuel, ces prédicats peuvent être pris comme des *faits* et tout ce que nous avons besoin de faire est de vérifier que les éléments dans les ensembles de restrictions de chaque variable globale sont unifiables ; chaque relation rhétorique est ainsi représentée par seulement *une* règle de dérivation (avec les ensembles de restrictions associés aux variables globales qui apparaissent dans cette règle) dans la grammaire abstraite d'attributs du discours entier ; il s'agit de la règle de dérivation qui correspond à la règle Prolog qui implémente la sémantique de cette relation rhétorique. Par conséquent, à l'étape 2 dans la procédure décrite ci-dessus, seules les règles (1) avec les ensembles d'attributs (1') sont pertinentes ; les règles et les ensembles d'attributs de (2) à (K), respectivement de (2') à (K') ne sont pas pris en compte.

Jusqu'à maintenant nous n'avons rien dit sur le rôle des règles de dérivation dans le filtrage des relations rhétoriques. En fait, ces règles jouent un rôle mineur dans le processus de filtrage ; elles peuvent servir seulement comme un moyen de dépanner l'utilisation des sémantiques des relations rhétoriques : si plusieurs règles de dérivation ont des têtes identiques, alors elles doivent aussi avoir des corps identiques, puisque chaque règle est une abstraction de la clause Prolog qui implémente la sémantique de la relation rhétorique, ainsi chaque tête d'une règle représente un type de relation rhétorique. Par conséquent, les corps devraient représenter les sémantiques des relations rhétoriques.

Une extension clé apportée par notre cadre aux travaux antérieurs concernant la conversion des programmes logiques en grammaires abstraites d'attributs [83] consiste dans la notion de *variable globale* dans un programme logique. Dans la description du filtrage contextuel des relations rhétoriques nous sommes passés assez vite sur cette notion, mais nous précisons ici qu'en faisant « globales » certaines variables, afin qu'elles gardent leur signification et leur valeur d'une règle à l'autre, nous ouvrons la voie vers une formalisation d'une consistance « syntaxique » de la grammaire. Cela est vrai parce que chaque élément dans un ensemble de restrictions représente en fait un *chemin* vers cette variable ; et si la variable garde sa valeur d'une

règle à l'autre, alors les éléments dans ses ensembles de restrictions devraient être unifiables. Ensuite, en prenant les variables globales pour les énoncés du discours, l'unifiabilité de leurs ensembles de restrictions associés revient essentiellement à ne pas avoir d'assertions contradictoires concernant ces énoncés.

Par exemple, nous ne pouvons pas avoir en même temps des prédicats tels que : $\text{equals}(\chi, \text{topic}(\alpha))$ et $\text{equals}(\chi, \text{SARG}(\alpha))$ pour les variables globales χ et α , car ces prédicats engendrent les « chemins » suivants vers α dans les ensembles de restrictions : $\lambda_2^{\text{equals}} \circ \text{id}(\text{topic}(*))$ et $\lambda_2^{\text{equals}} \circ \text{id}(\text{SARG}(*))$ respectivement, où $*$ marque la « profondeur » des prédicats dans les règles qui les contiennent ; du fait que ces éléments ne sont *pas* pertinents dans le processus d'unification, nous les avons substitués par ce symbole. L'unification entre les deux « chemins » n'est pas possible car ils sont équivalents à $\text{topic}(\lambda_2^{\text{equals}})$ et $\text{SARG}(\lambda_2^{\text{equals}})$ et ces deux expressions ne peuvent pas avoir la même valeur, puisque $\text{topic}/1$ retourne un *ensemble* de variables initialisées (dans l'énoncé pris comme argument), tandis que $\text{SARG}/1$ retourne une *clause* qui contient le type de l'acte de langage pour l'énoncé pris comme argument, avec l'action communiquée dans l'énoncé. Un problème est que si, pour l'exemple considéré ici, χ n'est plus une variable globale alors, bien que l'ensemble des restrictions pour α reste le même, il est parfaitement légitime que $\text{topic}(\alpha)$ s'unifie avec une instance de la variable locale χ , tandis que $\text{SARG}(\alpha)$ s'unifie avec une autre instance de la variable locale χ , dans une autre règle. La réponse est que, lorsque χ est locale, étant donné que le prédicat $\text{equals}/2$ ne fait que rebaptiser le prédicat $\text{Prolog} == /2$, $\lambda_2^{\text{equals}}$ est *par défaut* liée à $\lambda_1^{\text{equals}}$ qui est, à son tour, locale ; alors, il n'y a plus aucune contrainte pour construire l'ensemble de restrictions pour α .

Enfin, nous devrions remarquer que pour les relations rhétoriques dont les sémantiques contiennent le connecteur \vee (dont le programme logique correspondant contient « ; » et la grammaire abstraite contient « | »), les ensembles d'attributs calculés (et censés être unifiés) ne correspondent *qu'*aux « branches » dans la disjonction qui ont été effectivement trouvées valides dans le processus de résolution (lors du filtrage sémantique).

4.3.2 Fondements théoriques

Dans ce paragraphe nous allons montrer que le filtrage contextuel des relations rhétoriques est formellement correct, c'est-à-dire, qu'il est correct et complet. Autrement dit, nous allons montrer que toute structure rhétorique qui est cohérente (selon la Définition 1) a une grammaire abstraite d'attributs qui est « unifiable » (c'est-à-dire, dont tous les ensembles de restrictions associés à chacune des variables globales peuvent être unifiés) – la complétude, et que chaque structure rhétorique qui a une grammaire abstraite d'attributs « unifiable » associée est cohérente. Afin de prouver la correction théorique du filtrage contextuel, nous avons d'abord besoin d'un lemme :

Lemme 1. *Pour qu'une structure (segmentée) de discours soit cohérente, il est nécessaire que le programme logique obtenu en concaténant les sémantiques des relations rhétoriques, selon la structure de discours, soit consistant.*

Démonstration. Nous démontrons le lemme en supposant le contraire, c'est-à-dire qu'il existe une structure rhétorique cohérente telle que son programme logique correspondant n'est pas consistant.

Si ce programme logique n'est pas consistant, alors il ne peut réussir, donc la structure de discours ne peut être calculée comme une conséquence du programme.

Mais cela contredit la Définition 1 et la supposition que la structure de discours est cohérente. □

Ce lemme est utile pour prouver le résultat principal concernant le filtrage contextuel :

Proposition 1. *La procédure de filtrage contextuel des relations rhétoriques est correcte et complète.*

Démonstration. On prouve d'abord la complétude (c'est-à-dire, qu'à toute structure de discours cohérente correspond une grammaire abstraite d'attributs « unifiable »).

Pour ce faire, nous supposons le contraire, qu'il existe une structure rhétorique cohérente qui a une grammaire abstraite d'attributs qui contient une variable globale dont les ensembles de restrictions ne peuvent pas être unifiés. Dans ce cas, il résulte de la manière dont les grammaires abstraites d'attributs sont construites à partir des programmes logiques (et du Lemme 2 – voir ci-dessous) que le programme logique à l'origine de la grammaire n'est pas consistant.

On a donc une structure rhétorique cohérente qui a un programme logique inconsistant associé, ce qui contredit le Lemme 1.

Pour prouver la correction (c'est-à-dire, que toute structure rhétorique qui a une grammaire abstraite d'attributs « unifiable » est cohérente), nous supposons le contraire, qu'il existe une structure de discours non-cohérente avec une grammaire abstraite d'attributs qui est « unifiable ».

De la Définition 1, cela signifie que cette structure rhétorique n'est pas calculable *via* un programme logique, en utilisant les sémantiques des relations rhétoriques. Mais, de l'autre côté, la grammaire abstraite d'attributs qui correspond à cette structure rhétorique est « unifiable », correspondant ainsi à un programme logique qui est consistant, pourvu que les règles dans la grammaire soient consistantes (ce qui est vrai si les sémantiques des relations rhétoriques ont été définies de la manière précisée dans le chapitre 3, et ces dernières sont utilisées correctement).

On a donc un programme logique consistant qui, pourtant, ne réussit pas pour une attribution particulière de valeurs à ses variables globales (c'est-à-dire, aux énoncés). Cela signifie qu'il y a au moins une règle (Prolog) dans ce programme qui ne réussit pas, ce qui revient à l'existence d'au moins un prédicat appliqué à une variable globale qui (pour une valeur particulière de cette variable) échoue. Cela implique que la relation rhétorique dont la sémantique contient ce prédicat ne peut être vraie pour la valeur particulière de la variable concernée.

D'autre part, cette relation rhétorique était correcte au sens des filtrages locaux préalables, donc cette relation rhétorique est vraie pour la valeur particulière de la variable concernée, ce qui aboutit à une contradiction.

En conclusion, le fait que le filtrage local sur des bases sémantiques précède le filtrage global assure la correction de ce dernier. □

La preuve de la Proposition 1 s'est appuyée de manière implicite sur le fait qu'il y avait une correspondance surjective de l'ensemble des programmes logiques à l'ensemble des grammaires abstraites d'attributs, c'est-à-dire, que pour tout programme logique il existe une unique grammaire abstraite d'attributs (le caractère fonctionnel) et que plusieurs programmes logiques

peuvent mener à la même grammaire abstraite d'attributs (la surjectivité). De cette correspondance surjective nous avons qu'une grammaire abstraite d'attributs peut être regardée comme une *classe d'équivalence* (modulo le partage des variables en deux classes : « héritées » et « synthétisées ») pour un ensemble de programmes logiques. A partir de cette idée, nous pouvons déduire que la non-unifiabilité d'une série d'ensembles de restrictions pour une variable globale dans une grammaire abstraite d'attributs détermine l'inconsistance de tous les programmes logiques dans la même classe d'équivalence induite par la grammaire abstraite d'attributs.

Il faut donc prouver que la correspondance entre les programmes logiques et les grammaires abstraites d'attributs est une fonction surjective, ce qui est le résultat du :

Lemme 2. *La correspondance entre les programmes logiques et les grammaires abstraites d'attributs est une fonction surjective.*

Démonstration. On peut supposer, sans restreindre la généralité du problème, qu'un programme logique consiste en une clause de la forme :

$$q^0(q_1^1(q_1^2(\dots(q_1^D(X_1, \dots, X_{N_{D_1}})\dots)\dots)\dots)q_N^1(\dots q_M^1(\dots q_P^E(X_1, \dots, X_{M_{E_P}})\dots)\dots)\dots) : - \\ p_1^0(1p_1^1(1p_1^2(\dots(1p_1^F(X_1, \dots, X_{N_{F_1}})\dots)\dots)\dots)\dots)\mu(\xi_1)\dots\mu(\xi_{Q-1})p_Q^0(Qp_1^1(\dots Qp_M^2(\dots(\dots Qp_P^G(X_1, \dots, X_{M_{G_P}})\dots)\dots)\dots)\dots).$$

Dans cette formule, les indices supérieurs 0, 1, 2, ..., D, E, F, G désignent les profondeurs d'imbrication des prédicats et des variables, tandis que X_i désignent les variables dans le programme.

A partir de cette règle Prolog nous pouvons calculer une grammaire abstraite d'attributs, qui contient une règle et une série d'ensembles de restrictions, un ensemble de restrictions pour chaque variable. Donc, ces éléments sont montrés ci-dessous :

(i) la règle : $q^0 \rightarrow p_1^0\tau(\mu(\xi_1))\dots\tau(\mu(\xi_{Q-1}))p_Q^0$;

(ii) les ensembles de restrictions pour les variables (il existe les entiers T et R qui satisfont les expressions suivantes et $i \in \{1, \dots, \max(N_{D_1}, \dots, M_{E_P}, \dots, N_{F_1}, \dots, M_{G_P})\}$) : $X_i : \{\lambda_1^{q^0} \circ \lambda_1^{q^1} \circ \dots \circ \lambda_1^{q^{D-1}} \circ id(q_1^D.i(\epsilon)), \dots, \lambda_N^{q^0} \circ \lambda_M^{q^1} \circ \dots \circ \lambda_T^{q^{E-1}} \circ id(q_P^E.i(\epsilon)), \lambda_1^{p_1^0} \circ \lambda_1^{p_1^1} \circ \dots \circ \lambda_1^{p_1^{F-1}} \circ id(1p_1^F.i(1)), \dots, \lambda_1^{p_1^0} \circ \lambda_M^{p_1^2} \circ \dots \circ \lambda_R^{p_1^{G-1}} \circ id(Qp_P^G.i(Q))\}$.

Dans cette construction, il est évident que, structurellement, la grammaire abstraite d'attributs est déterminée de manière unique par la règle et par la famille des ensembles de restrictions ; par conséquent, pour le programme logique donné, elle est unique. Nous avons ainsi démontré la première partie du lemme, le fait que la correspondance des programmes logiques aux grammaires abstraites d'attributs est une fonction.

Pour prouver que cette fonction est surjective, nous avons besoin de montrer qu'il peut y avoir au moins deux programmes logiques qui ont la même grammaire abstraite d'attributs et que pour tout programme logique il existe une grammaire. La dernière partie de cette assertion est évidente, du fait que la grammaire abstraite d'attributs est construite mécaniquement à partir du programme logique. Pour la première partie de l'assertion, soient deux programmes logiques qui consistent, chacun, dans une règle du type montré ci-dessus. Pour alléger les notations, nous supposons que les programmes sont LP et \overline{LP} , où LP est précisément comme montré ci-dessus, et \overline{LP} a la forme :

$$\overline{q^0}(\overline{q_1^1}(\overline{q_1^2}(\dots(\overline{q_1^D}(X_1, \dots, X_{\overline{N_{D_1}})}\dots)\dots)\dots)\dots)\overline{q_N^1}(\dots\overline{q_M^1}(\dots\overline{q_P^E}(X_1, \dots, X_{\overline{M_{E_P}})}\dots)\dots)\dots) : -$$

$\overline{p_1^0}(\overline{1p_1^1}(\overline{1p_1^2}(\dots(\overline{1p_1^F}(X_1, \dots, X_{N_{F_1}})\dots)\dots)\dots)\dots)\mu(\overline{\xi_1})\dots\mu(\overline{\xi_{Q-1}})\overline{p_Q^0}(\overline{Qp_1^1}(\dots\overline{Qp_M^2}(\dots(\dots\overline{Qp_P^G}(X_1, \dots, X_{M_{G_P}})\dots)\dots)\dots)\dots)$.

La grammaire abstraite d'attributs correspondante, \overline{AAG} contient :

(i) la règle : $\overline{q^0} \rightarrow \overline{p_1^0}\tau(\overline{\mu(\xi_1)})\dots\tau(\overline{\mu(\xi_{Q-1})})\overline{p_Q^0}$;

(ii) les ensembles de restrictions des variables : $X_i : \{\lambda_1^{q^0} \circ \lambda_1^{q_1^1} \circ \dots \circ \lambda_1^{q_1^{p-1}} \circ id(\overline{q_1^D}.i(\epsilon)), \dots, \lambda_N^{q^0} \circ \lambda_M^{q_M^1} \circ \dots \circ \lambda_T^{q_T^{E-1}} \circ id(\overline{q_P^E}.i(\epsilon)), \lambda_1^{p_1^0} \circ \lambda_1^{p_1^1} \circ \dots \circ \lambda_1^{p_1^{F-1}} \circ id(\overline{1p_1^F}.i(1)), \dots, \lambda_1^{Qp_1^1} \circ \lambda_M^{Qp_M^2} \circ \dots \circ \lambda_R^{Qp_R^{G-1}} \circ id(\overline{Qp_P^G}.i(\overline{Q}))\}$.

Les deux grammaires, AAG (pour LP) et \overline{AAG} (pour \overline{LP}) sont identiques si et seulement si leurs règles sont identiques *et* leurs ensembles d'attributs peuvent être unifiés, pour chaque variable. L'identité des règles, dans ce cas, revient à l'identité des foncteurs et des arités pour les prédicats de profondeur 0, et à l'identité des connecteurs, c'est-à-dire, par exemple, en adoptant une syntaxe à la Prolog, $\text{functor}(\overline{q^0}, _f, _a)$ et $\text{functor}(q^0, _f, _a)$ sont vraies simultanément pour les mêmes liens (« bindings ») pour $_f$ et $_a$, et $\overline{\xi_i} \equiv \xi_i$, pour i de 1 à $Q - 1$. L'unifiabilité des ensembles d'attributs revient à l'unifiabilité de chaque paire de termes dans la réunion des ensembles d'attributs pour chaque variable et pour chacun des deux programmes. Cependant, la construction de chaque grammaire abstraite d'attributs et la supposition que chaque programme logique est consistant assurent que les cas intéressants d'unification concernent des paires d'éléments tels que chacun est dans un ensemble de restrictions différent, pour une variable. Plus précisément, en notant par $RS(X_i)$ l'ensemble des restrictions pour la variable X_i dans la grammaire AAG et par $\overline{RS}(X_i)$ l'ensemble des restrictions de la même variable dans la grammaire \overline{AAG} , nous devrions étudier seulement les équations de la forme $Y = Z$, où $Y \in RS(X_i)$ et $Z \in \overline{RS}(X_i)$. Par exemple, nous devrions avoir que⁴ :

$\lambda_N^{q^0} \circ \lambda_M^{q_M^1} \circ \dots \circ \lambda_T^{q_T^{E-1}} \circ id(\overline{q_P^E}.i(\epsilon)) = \lambda_N^{q^0} \circ \lambda_M^{q_M^1} \circ \dots \circ \lambda_T^{q_T^{E-1}} \circ id(\overline{q_P^E}.i(\epsilon))$, ce qui est équivalent à :

$$q_P^E.i(\lambda_T^{q_T^{E-1}}(\dots(\lambda_M^{q_M^1}(\lambda_N^{q^0}(\epsilon))))\dots) = \overline{q_P^E}.i(\lambda_T^{q_T^{E-1}}(\dots(\lambda_M^{q_M^1}(\lambda_N^{q^0}(\epsilon))))\dots).$$

Il est donc seulement nécessaire que les compositions de plusieurs fonctions soient identiques ; cela n'implique pas forcément que toutes les fonctions dans la composition soient respectivement identiques. Si toutes les fonctions dans les compositions étaient respectivement identiques, alors les deux programmes logiques seraient identiques ; du fait que cela n'est pas le cas, nous avons une classe de programmes qui mènent à la même grammaire abstraite d'attributs. \square

Le Lemme 2 ouvre la voie vers la recherche des conditions qu'un ensemble de programmes logiques devraient accomplir afin qu'ils soient dans la même classe d'équivalence, c'est-à-dire, qu'ils mènent à la même grammaire abstraite d'attributs. Dans la littérature [83] on affirme en général qu'une grammaire abstraite d'attributs représente une classe de programmes équivalents au niveau syntaxique, mais avec tous les regroupements possibles des variables en classes – « héritées » et « synthétisées ». Evidemment, cela est vrai, comme nous venons de le voir de

⁴En fait, de la manière dont les grammaires abstraites d'attributs sont déterminées et du présupposé que les deux programmes logiques sont consistants, il suffit de montrer que nous pouvons unifier seulement *une* paire d'éléments dans les ensembles de restrictions qui appartiennent aux différentes grammaires.

la démonstration du Lemme 2, mais l'identité des programmes logiques, à l'exception des regroupements des variables (en d'entrée / de sortie) est une condition suffisante et non-nécessaire qu'un ensemble de programmes logiques mènent à la même grammaire abstraite d'attributs. Il serait donc intéressant de trouver des conditions nécessaires que deux programmes logiques mènent à la même grammaire abstraite d'attributs, car cela nous permettrait d'établir des classes d'équivalence des structures de discours, pour ainsi décider plus rapidement sur la cohérence de ces structures.

4.3.3 Discussion

Le filtrage contextuel des relations rhétoriques pour la mise à jour des structures de discours a été décrit en détail et sa pertinence théorique discutée. Cependant, il y a quelques aspects qui méritent une discussion plus approfondie.

Tout d'abord, nous devrions étudier ce qui se passe dans le processus de filtrage contextuel lorsqu'il n'y a pas d'historique de dialogue disponible. Plus précisément, il s'agit de voir l'effet des contraintes contextuelles sur les relations rhétoriques entre deux énoncés, lorsque ces derniers sont les deux premiers en dialogue. La réponse (évidente) est que le filtrage contextuel n'a aucun effet dans ce cas, car (i) le contexte est réduit aux deux énoncés en question, (ii) la seule règle dans chaque grammaire abstraite d'attributs au niveau du discours contient chaque relation rhétorique qui était passée par le filtrage local (pragmatique et sémantique), donc les ensembles d'attributs des variables « globales » concernées sont, par nécessité, unifiables.

De toute manière, une réponse moins évidente à cette question est que le filtrage contextuel peut jouer un rôle de dépannage : lorsque nous trouvons des éléments non-unifiables dans les ensembles de restrictions des deux variables globales, cela signifie que le filtrage sémantique ne s'est pas déroulé correctement ; en supposant que la correction du processus inférentiel est assurée (par exemple en utilisant Prolog), l'incorrection du processus de filtrage sémantique peut montrer des bogues dans la définition des prédicats discursifs, dans l'ontologie de tâches, ou dans les deux. Pourtant, étant donné que les prédicats discursifs sont indépendants du domaine particulier et ont ainsi été déjà validés par l'usage dans plusieurs domaines, nous pouvons conclure que le filtrage contextuel peut servir à dépanner l'ontologie de tâches, particulière pour l'application considérée.

De plus, du fait que lorsqu'aucun contexte dialogique n'est disponible, le filtrage contextuel n'a aucun effet sur l'ensemble des relations rhétoriques pertinentes entre une paire d'énoncés, les deux premiers énoncés en dialogue sont connectés essentiellement (sauf s'il y a des bogues dans l'ontologie de tâches – voir ci-dessus) *via* les relations rhétoriques passées par les filtres pragmatique et sémantique, donc, *via* une ou *plusieurs* relations rhétoriques.

Un deuxième problème, plus subtil, sur le filtrage contextuel, concerne une autre situation « limite » dans la mise à jour des structures rhétoriques, où les filtres pragmatique et sémantique ont fourni au moins une relation rhétorique valide entre une paire d'énoncés (ou constituants discursifs), le filtrage contextuel ne sort aucune relation rhétorique entre ces énoncés et, par conséquent, le dernier énoncé dans le discours n'est connecté à aucun énoncé antérieur.

Dans ce cas, si nous adoptons l'heuristique de pénaliser les structures de discours plus « courtes » (c'est-à-dire, avec un nombre plus bas d'énoncés) en privilégiant les structures plus « longues », alors, si un « hiatus » apparaît dans la structure rhétorique (c'est-à-dire, il y a un

énoncé qui n'est relié à aucun énoncé antérieur dans le même discours), nous pouvons faire *un retour en arrière* vers le « point » antérieur dans le discours (ici, par « point » dans le discours nous entendons une paire d'énoncés ou constituants discursifs connectés par des relations rhétoriques) où plusieurs relations rhétoriques sont sorties des trois filtres – pragmatique, sémantique *et* contextuel.

La relation rhétorique couramment choisie dans ce « point » dans le discours (avant le retour en arrière) est celle avec le coût computationnel le plus réduit (c'est-à-dire, qui a passé le filtrage numérique – voir ci-dessous); maintenant, si nous avons besoin de faire un retour en arrière (donc, il y a un « hiatus »), alors, dans ce « point »-là dans le discours où plusieurs relations rhétoriques alternatives étaient disponibles, nous choisissons la relation rhétorique avec le coût computationnel le plus réduit, sauf la relation antérieurement choisie et nous *re-appliquons* le filtrage contextuel sur tous les énoncés à partir du « point » en question, jusqu'à l'énoncé où le « hiatus » était apparu.

Un exemple de dialogue avec hiatus, entre deux locuteurs *U* et *M* dans le contexte d'une tâche de réservation de livres dans une bibliothèque, est (par : nous avons marqué le « hiatus ») :

M : Bonjour, qu'est-ce que je peux faire pour vous ?

U : Eh ben, je voudrais un bouquin sur l'astronomie...

M : Vous voulez un livre de niveau avancé, ou plutôt une introduction au domaine ?

:

U : Il fait beau sur Mars ?

Il y a trois possibilités pour traiter ces situations de « hiatus » :

- (i) le dernier énoncé qui apparaît dans le discours est relié à quelques-uns des énoncés antérieurs et le « hiatus » est ainsi éliminé ;
- (ii) le « hiatus » est maintenu au même « point » dans le discours, c'est-à-dire, au point qui contient le dernier énoncé censé être relié à la structure rhétorique ;
- (iii) un autre « hiatus » apparaît plus tôt dans le discours, c'est-à-dire, au niveau d'un énoncé produit avant le dernier énoncé dans la structure de discours entière (où le « hiatus » était apparu), mais après le dernier énoncé dans le « point » du discours vers lequel nous avons effectué le retour en arrière.

La deuxième situation montre qu'effectuer un retour en arrière et modifier un « point » de discours n'apporte aucun avantage au processus de mise à jour de la structure rhétorique, tandis que la troisième situation montre que le retour en arrière dégrade les structures rhétoriques obtenues, en les rendant encore plus courtes. Ainsi, si la situation (ii) apparaît, alors nous gardons les modifications réalisées à ce « point »-là dans le discours, et nous effectuons encore un retour en arrière vers un « point » antérieur dans le discours, où plusieurs relations rhétoriques alternatives avaient passé le filtrage contextuel, et dans ce « point » nous changeons la relation rhétorique, pour celle dont le coût computationnel est le moins élevé (sauf la relation qui était déjà en place), ensuite nous re-appliquons le filtrage contextuel au reste des énoncés et des relations rhétoriques à partir de ce « point », et ainsi de suite.

Si nous nous trouvons en situation (iii), alors nous *supprimons* la modification faite au dernier « point » dans le discours, avant le dernier énoncé, et soit nous essayons une autre relation rhétorique alternative au même « point » et re-applique le filtrage contextuel à partir de ce « point », soit, s'il n'y a plus aucune relation rhétorique alternative, nous effectuons un retour en arrière vers un autre « point » dans le discours, plus ancien, où il existe des relations rhétoriques alternatives et nous procédons comme en situation (ii). Si la situation (i) apparaît, alors, évidemment, cela est ce que nous voulons et nous pouvons donc continuer le processus de mise à jour de la structure rhétorique.

Pour un traitement plus précis de cette stratégie de « hiatus coping », nous supposons que dans une structure rhétorique il y a plusieurs énoncés, libellés π_i , où i varie de 1 à N , avec π_N l'énoncé le plus récent. Ensuite, nous supposons qu'à l'exception de la paire initiale d'énoncés (où, nous le rappelons, le filtrage contextuel n'a, normalement, aucun effet sur les relations rhétoriques), il y a deux « points » dans le discours où des relations rhétoriques alternatives sont sorties avec succès des filtrages pragmatique, sémantique et contextuel. Nous allons noter ces « points » par les énoncés π_i et π_j (le premier « point ») et les énoncés π_l et π_m (pour le deuxième « point »). Nous supposons ensuite, sans restreindre la généralité du traitement, qu'à chacun de ces deux « points » dans le discours il y a deux relations rhétoriques alternatives qui sont passées avec succès par les trois filtrages (pragmatique, sémantique et contextuel) : ρ_k et ρ'_k pour le premier « point » dans le discours, et ρ_n et ρ'_n pour le deuxième « point ». De plus, nous allons marquer par $\rho...$ les relations rhétoriques dont les types ne sont pas pertinents pour le traitement, par N l'indice du dernier énoncé dans le discours, par Δ , un nombre entier positif supérieur ou égal à 2 tel que $N - \Delta > m$, et par ρ_M une relation rhétorique qui est, en fin de compte (en situation (i)) trouvée valide entre les énoncés π_{N-1} et π_N .

Avec ces notations, nous illustrons la stratégie de « hiatus coping » dans la figure 4.1, où nous avons marqué par des flèches en pointillée les relations rhétoriques alternatives non encore sélectionnées, par des flèches continues, les relations rhétoriques courantes, par des flèches continues coupées par une croix les relations rhétoriques précédemment sélectionnées mais juste supprimées, par les arcs non-fléchis accompagnés de « ? » nous entendons qu'il pourrait y avoir des relations rhétoriques entre les énoncés concernés, mais cela doit être calculé, et par les des flèches en ligne pointillée nous entendons des relations rhétoriques qui pourraient ne plus être valides, à cause des changements dans les relations rhétoriques aux « points » antérieurs dans le discours. Dans la figure 4.1 nous avons représenté ainsi la stratégie de « hiatus coping », tout en montrant la situation de « hiatus » initiale, deux étapes de retour en arrière, correspondant aux situations (ii) et (iii), et à la situation « idéale » (i). Nous observons que si, après avoir effectué un retour en arrière nous ne résolvons pas les « hiatus » et si des « points » antérieurs dans le discours, avec des relations rhétoriques alternatives, sont disponibles, alors d'autres retours en arrière ont lieu, pour essayer d'arriver à la situation (i).

Cependant, cette situation de « hiatus » (c'est-à-dire, où un énoncé qui fait, *a priori*, partie d'un dialogue ne peut être relié *via* des relations rhétoriques à aucun énoncé ou constituant discursif antérieur dans ce dialogue) n'apparaît que rarement en pratique ; dans la plupart des dialogues *réels* étudiés jusqu'à présent (des corpus construits *via* la méthode du magicien d'Oz, ou des pièces de théâtre), nous n'avons, en fait, trouvé aucun énoncé qui ne puisse être connecté *via* une relation rhétorique à des énoncés antérieurs dans le dialogue dont il fait partie. Il n'en est

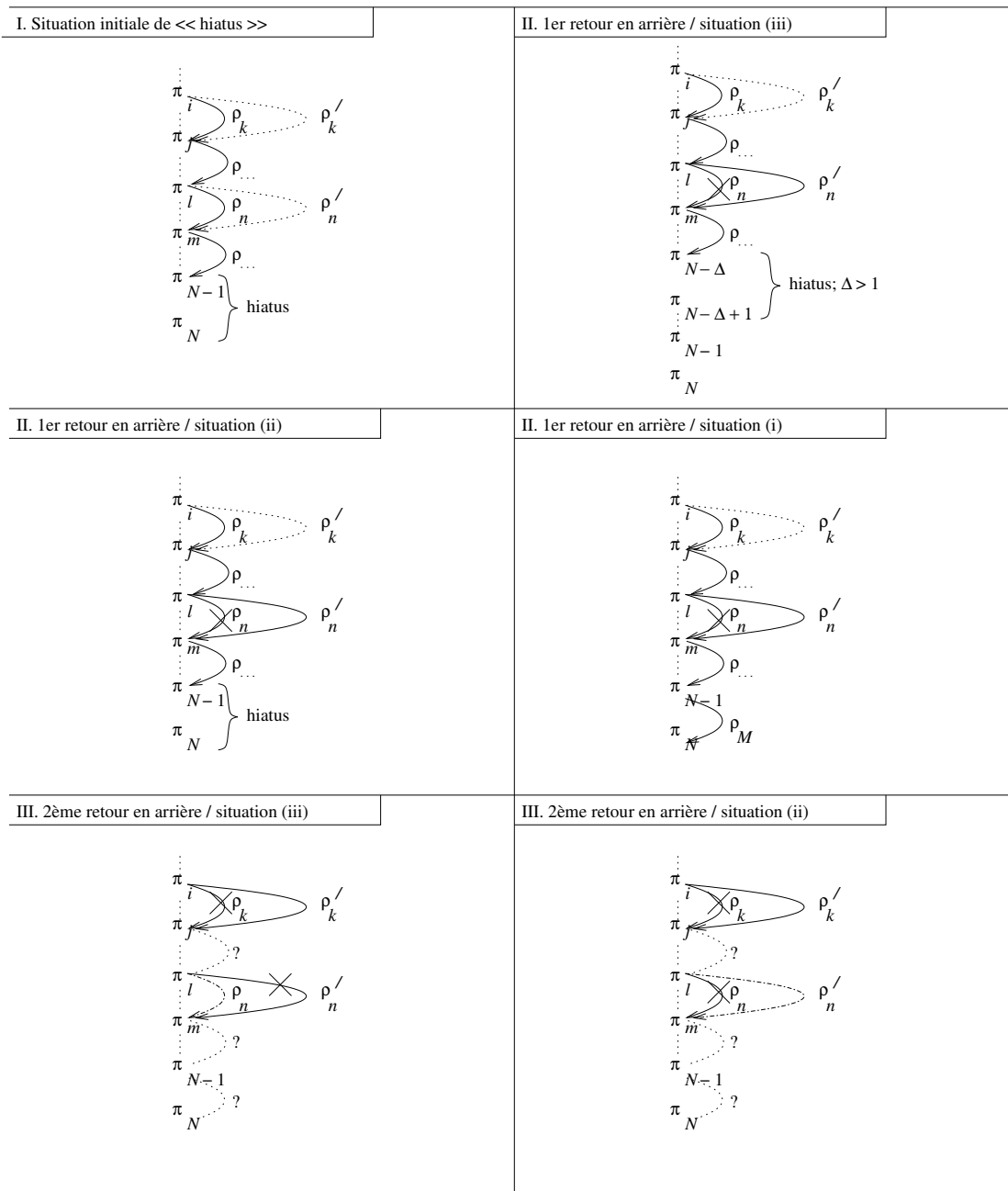


FIG. 4.1 – Stratégie de « coping » avec le « hiatus » discursif

pas moins vrai que la discussion menée ci-dessus reste importante lorsque les interlocuteurs produisent des énoncés très ambigus, ou des énoncés accompagnés par des éléments multimodaux (tels que des gestes), ou même des énoncés ironiques. Par exemple, prenons le dialogue suivant entre l'ordinateur *M* (qui joue le rôle d'une bibliothécaire) et un client *U* dans une bibliothèque :

U : Bonjour, est-ce que je pourrais avoir quelques bouquins en géométrie algébrique ?

M : Bonjour Monsieur, vous voulez un livre de niveau moyen ou avancé ?

U : Eh ben, je commencerais bien un bouquin de niveau moyen, et puis on verra...

M : OK, je peux vous donner le livre « Fondements de la géométrie algébrique » par Alexandre Rivière, il s'agit d'une introduction très accessible, bien que rigoureuse, à ce sujet.

U : * Eh ben, est-ce qu'on s'y noie souvent ?

Evidemment, le dernier énoncé du locuteur est difficile à interpréter, ainsi il est difficile de trouver de manière immédiate une relation rhétorique qui le relie au reste du dialogue, car il s'agit d'un jeu de mots : soit on se « noie » à cause de la difficulté de l'ouvrage, soit on se « noie » par rapport au nom de l'auteur 'Rivière'.

Enfin, nous soulignons un détail technique mineur mais toutefois important concernant la stratégie de « hiatus coping » décrite ci-dessus : lors du choix entre plusieurs relations rhétoriques dans un « point » de discours, nous utilisons le filtrage numérique (décrit ci-dessous, dans le paragraphe suivant) ; la manière précise dont nous faisons cela consiste à *trier* de manière croissante les relations rhétoriques alternatives à chaque « point » dans le discours, par rapport à un score computationnel, proportionnel au temps requis pour calculer chacune des relations rhétoriques. Ainsi, même si le filtrage numérique n'est pas très important *per se*, il aide à guider l'exploration des relations rhétoriques alternatives lors des retours en arrière vers des « points » dans le discours.

4.3.4 Filtrage numérique des relations rhétoriques

Pour avoir une heuristique qui permette de choisir entre plusieurs relations rhétoriques alternatives entre une paire de constituants discursifs ou d'énoncés, nous associons à chaque grammaire abstraite d'attributs obtenue à partir de la sémantique d'une relation rhétorique, un coût computationnel, proportionnel au temps nécessaire pour calculer cette relation entre deux énoncés (disponibles sous forme logique). En cas d'égalité de coûts computationnels, nous choisissons la relation rhétorique de la plus faible portée (c'est-à-dire, qui relie l'énoncé courant à un énoncé antérieur aussi récent que possible). D'un autre côté, l'exemple que nous avons discuté dans le paragraphe précédent montre qu'il faudrait garder les deux interprétations car il y a un double jeu de mots. Cependant, nous souhaitons garder une seule relation rhétorique pour des raisons computationnelles et du fait qu'on n'attend pas d'une machine de traiter des jeux de mots ni l'ironie.

Le filtrage numérique suppose une étape préalable où, pour une relation rhétorique donnée ρ , nous calculons les quantités suivantes :

- $|AAG_\rho| ::=$ nombre total de règles de production dans la grammaire associée à ρ ;
- $|AAG_\rho^{(N)}| ::=$ nombre total de règles de production qui contiennent des non-terminaux dans leur partie droite ;
- pour chaque règle (r) dans la grammaire abstraite d'attributs associée à ρ , $|Var_\rho(r)| ::=$ nombre total de variables différents (c'est-à-dire, d'ensembles d'attributs) dans la règle (r) ;

- pour chaque règle (r) dans la grammaire associée à ρ , et pour chaque variable différente $_v$ dans $Var_\rho(r)$, $|Attr_\rho^{(r)}(_v)| ::=$ taille (c'est-à-dire, nombre d'éléments) de l'ensemble de restrictions associé à la variable $_v$ dans la règle (r) dans la grammaire associée à la relation rhétorique ρ .

Ces quantités ne sont pas utilisées pour calculer une « valeur » absolue pour le coût, mais, au lieu de cela, permettent d'évaluer les charges *relatives*, en comparant deux ou plusieurs relations rhétoriques, appliquées au même couple d'énoncés (ou constituants discursifs). Dans la présentation qui suit, nous allons regarder de manière assez détaillée ce processus de comparaison. Ainsi, en considérant qu'il y a deux relations rhétoriques ρ et σ qui relie la même paire d'énoncés, la procédure pour comparer les coûts relatifs de calcul est la suivante :

(i) si $|AAG_\rho| = |AAG_\sigma|$, alors aller à l'étape suivante de comparaison ; sinon, choisir la relation avec le nombre minimal de règles de dérivation ;

(ii) si $\frac{|AAG_\rho^{(N)}|}{|AAG_\rho|} = \frac{|AAG_\sigma^{(N)}|}{|AAG_\sigma|}$, alors aller à l'étape suivante ; sinon, choisir la relation avec le rapport minimal entre le nombre des règles avec des non-terminaux dans la partie droite, et le nombre total de règles ; l'intuition derrière cette étape est que les règles avec des non-terminaux dans leur partie droite ont encore besoin de dérivations, afin d'arriver aux terminaux, pour ainsi les rendre capables de réussir ; ainsi, ces règles induisent un certain coût supplémentaire *futur* ;

(iii) si pour toutes les règles (r) dans AAG_ρ et (s) dans AAG_σ , nous avons que $|Var_\rho(r)| = |Var_\sigma(s)|$, alors aller à l'étape suivante ; sinon, choisir la relation rhétorique dont la grammaire AAG a la propriété que $\sum_{(r) \in AAG} |Var(r)|$ est minimale ; l'intuition derrière cela est que plus il y a de variables dans une règle, plus d'unifications doivent être réalisées, ce qui induit un coût supplémentaire ;

(iv) si pour toutes les règles (r) dans AAG_ρ et (s) dans AAG_σ et toutes les variables $_v$ dans $Var_\rho(r)$ et $_w$ dans $Var_\sigma(s)$, nous avons que $|Attr_\rho^{(r)}(_v)| = |Attr_\sigma^{(s)}(_w)|$, alors choisir de manière aléatoire une des deux relations rhétoriques ; sinon, choisir la relation dont la grammaire AAG a la propriété que $\sum_{(r) \in AAG} \sum_{_v \in Var(r)} |Attr_\rho^{(r)}(_v)|$ est minimale ; l'intuition derrière cela est que des ensembles de restrictions plus grands déterminent un nombre plus grand d'égalités (entre les membres de ces ensembles) à vérifier, dont un nombre plus important de processus d'unification, ce qui revient à un coût de calcul supplémentaire.

Les entrées dans la procédure qui compare les relations rhétoriques en termes de leurs coûts computationnels consistent en : (a) une paire d'énoncés (spécifiés sous forme logique), reliés *via* strictement plus d'une relation rhétorique ; (b) les types (noms) des relations rhétoriques qui relie cette paire d'énoncés⁵. La sortie de l'algorithme est représentée par strictement une relation rhétorique, contenue dans l'ensemble passé dans l'entrée (b).

Avant d'illustrer ce cadre sur un exemple de dialogue réel, nous devons préciser qu'en réalité il ne faudrait pas filtrer trop fort, pour maintenir les cas d'ambiguïté le plus loin possible avant résolution.

4.3.5 Exemple étendu en dialogue

Dans cette section nous allons illustrer de manière très détaillée le filtrage contextuel, *via* un exemple de dialogue réel, un fragment de la scène VI, acte I, du vaudeville d'Eugène Labiche,

⁵Ces noms des relations rhétoriques agissent comme des pointeurs vers les sémantiques des relations rhétoriques.

« La cagnotte »⁶. Blanche et Léonida sont les personnages de ce dialogue. L'analyse n'essayera que d'évaluer le filtrage contextuel ; la génération proprement dite des répliques est simulée, *via* les étapes suivantes :

(i) tout d'abord, le dialogue est annoté en termes d'étiquettes pour les énoncés : à chaque énoncé nous associons une étiquette de la forme π_i^j , où i est l'indice du tour de parole courant et j , l'indice de l'énoncé courant dans ce tour de parole ;

(ii) ensuite, un annotateur humain crée les formes logiques pour les énoncés (il faut tenir compte du fait que certains énoncés sont elliptiques, donc leur sémantique complète doit être récupérée au niveau logique) ;

(iii) ces énoncés sont mis dans une file et placés en entrée des filtres pragmatique et sémantique, un énoncé à la fois ; ces filtrages sont donc appliqués à plusieurs reprises, pour mettre à jour la structure rhétorique du dialogue, avec chaque tour de parole : en fait, les énoncés dans un tour de parole sont mis en entrée des filtrages locaux tous à la fois, et ce sont les tours mêmes qui sont séquentiels dans le temps ;

(iv) après chaque filtrage pragmatique et sémantique en cascade, nous appliquons, en fin de compte, le filtrage contextuel ;

(v) si, après l'étape (v), il existe une paire d'énoncés avec au moins deux relations rhétoriques qui les relient, alors le filtrage numérique est appliqué, afin de supprimer certaines de ces relations, pour que chaque paire d'énoncé ne soit connectée que par (au plus) une relation rhétorique.

Pour les buts de ce chapitre, les étapes de (i) à (iii) ne seront pas détaillées ici ; seuls les résultats obtenus aux étapes (iv) et (v) seront mis en évidence. Ainsi, le dialogue est présenté ci-dessous :

Blanche : Ô, ma tante ! ^{π_1^1} Si tu savais comme je suis contente ! ^{π_1^2}

Léonida : En effet... ^{π_2^1}

Blanche : M. Félix vient de demander ma main à papa ^{π_3^1} et papa lui dit d'espérer... ^{π_3^2}

Léonida : Comment, tu aimes M. Félix ? ^{π_4^1}

Nous allons présenter ci-dessous une trace sur le processus de mise à jour de la structure rhétorique, au fur et à mesure que chaque tour de parole devient disponible aux algorithmes. Pour ce faire, nous « modélisons » les deux personnages par la machine, au sens où la structuration rhétorique est réalisée pour des tours de parole qui proviennent des deux personnages, Blanche et Léonida. Ainsi, nous allons montrer comment la mise à jour de la structure rhétorique est réalisée de manière dynamique, en unifiant les processus d'interprétation [193] et de structuration rhétorique pour la génération, à travers : 1) les types d'actes de langage réalisés par les énoncés, 2) les relations rhétoriques autorisées par les paires d'actes de langage, 3) les relations rhétoriques ayant passé le filtrage sémantique, et 4) les *détails* du filtrage contextuel.

Tout d'abord, Blanche produit un tour de parole composé de deux énoncés, π_1^1 et π_1^2 , tous les deux réalisant des actes du type F^S (« faire-savoir ») et étant produits par le même locuteur. Par conséquent, les relations rhétoriques monologiques qui sont autorisées par cette paire d'actes de langage (en passant avec succès par le filtrage pragmatique) sont *Elaboration*, *Contrast*,

⁶La version électronique de la pièce a été téléchargée de <http://fr.wikisource.org>.

Consequence et *Narration*. De son côté, le filtrage sémantique ne retient que les relations *Elaboration*(π_1^1, π_1^2) et *Consequence*(π_1^1, π_1^2). Théoriquement, le filtrage contextuel ne devrait imposer aucune contrainte supplémentaire sur les deux relations rhétoriques qui relient ces deux premiers énoncés, parce qu'il n'y a pas d'énoncés antérieurs pour composer un historique du dialogue. Néanmoins, le filtrage contextuel sera appliqué et son fonctionnement explicité ci-après, à titre d'exemple pour le lecteur, mais aussi afin de détecter des bogues possibles dans le filtrage sémantique. Ainsi, les ensembles de restrictions pour π_1^1 et π_1^2 sont, pour *Elaboration*⁷ :

$$\begin{aligned} & \text{Elaboration}(\pi_1^1, \pi_1^2) : \\ & \pi_1^1 : \{id(\text{Elab.1}(\epsilon)), id(\text{enounce}(1)), \lambda_2^{\text{SubclassOf}} \circ id(\text{topic}(3))\}; \\ & \pi_1^2 : \{id(\text{Elab.2}(\epsilon)), id(\text{enounce}(2)), \lambda_1^{\text{SubclassOf}} \circ id(\text{topic}(3))\}. \end{aligned}$$

On peut observer que tous les éléments dans chaque ensemble de restrictions sont unifiables, donc le filtrage sémantique s'est déroulé correctement. Pour la relation de *Consequence* entre ces énoncés, les ensembles de restrictions sont :

$$\begin{aligned} & \text{Consequence}(\pi_1^1, \pi_1^2) : \\ & \pi_1^1 : \{id(\text{Consequence.1}(\epsilon)), id(\text{enounce}(1)), \lambda^\neg \circ id(K(3))\}^8 ; \\ & \pi_1^2 : \{id(\text{Consequence.2}(\epsilon)), id(\text{enounce}(2)), id(K(3))\}. \end{aligned}$$

On peut voir qu'aucun bogue n'est signalé ici non plus, donc la relation *Consequence* est valide elle aussi.

Deuxièmement, Léonida génère l'énoncé π_2^1 , qui réalise un acte F^S . Ainsi, le filtrage pragmatique sort en tant que relations *dialogiques* (car les émetteurs des énoncés sont différents) potentiellement valides entre π_1^1 et π_2^1 , et entre π_1^2 et π_2^1 aussi, les relations suivantes : *ACK*, *P-Corr* et *P-Elab*. Ensuite, le filtrage sémantique calcule comme valides seulement les relations *ACK*(π_1^1, π_2^1) et *ACK*(π_1^2, π_2^1). Ces relations doivent maintenant être validées par le filtrage contextuel ; elles engendrent de nouveaux ensembles de restrictions pour π_1^1, π_1^2 et π_2^1 :

$$\begin{aligned} & \text{ACK}(\pi_1^1, \pi_2^1) \text{ et } \text{ACK}(\pi_1^2, \pi_2^1) : \\ & \pi_1^1 \text{ et } \pi_1^2 : \{id(\text{ACK.1}(\epsilon)), id(\text{enounce}(1)), \lambda_1^{\text{Disjoint}} \circ id(\text{topic}(3)), \lambda_1^{\text{confirmation}} \circ id(\text{SARG}(4))\}; \\ & \pi_2^1 : \{id(\text{ACK.2}(\epsilon)), id(\text{enounce}(2)), \lambda_2^{\text{Disjoint}} \circ id(\text{topic}(3)), id(\text{confirmation.2}(4))\}. \end{aligned}$$

Donc, en supposant tout d'abord que *Elaboration* est vraie entre π_1^1 et π_2^1 , nous observons que, du fait que π_1^1 et π_2^1 ont le même ensemble de restrictions que celui déterminé par la relation de *ACK* pour π_2^1 mais, de l'autre côté, l'ensemble de restrictions de ces deux énoncés devrait s'unifier avec les ensembles de restrictions tels que déterminés dans la mise à jour précédente de la structure rhétorique, c'est-à-dire, par la relation de *Elaboration* entre eux. Cela implique le fait que les ensembles de restrictions pour ces énoncés, tels que déterminés par la *Elaboration* entre eux, devrait être unifiables, donc, $\lambda_1^{\text{SubclassOf}} \circ id(\text{topic}(3))$ devrait s'unifier avec $\lambda_2^{\text{SubclassOf}} \circ id(\text{topic}(3))$, ce qui revient au fait que les *deux* arguments du prédicat *SubclassOf/2* devraient être identiques, ce qui contredit la sémantique de ce prédicat qui, pourtant, a été validé par le filtrage sémantique. Par conséquent, au moins une des relations *ACK*(π_1^1, π_2^1) et *ACK*(π_1^2, π_2^1) n'est pas valide. Nous choisissons n'importe quelle relation pour assurer la cohérence globale (dans le sens de notre théorie) de la structure rhétorique courante (parce que les ensembles

⁷Les règles dans la grammaire abstraite d'attributs ne sont pas explicitées, car nous supposons que les sémantiques des relations rhétoriques sont utilisées de manière consistante, c'est-à-dire, nous n'associons pas plusieurs expressions sémantiques à la même relation rhétorique.

⁸Tout au long de ce chapitre, λ^\neg a la même signification que λ^{neg} .

de restrictions deviennent unifiables – il n’y a aucun problème à unifier les éléments dans les ensembles de restrictions de n’importe quel énoncé de π_1^1 et π_1^2 tels que déterminés par la relation *Elaboration* entre eux, avec n’importe quel ensemble de restrictions pour les énoncés π_1^1 et π_1^2 , tels que déterminés par la relation de *ACK* entre n’importe quel de ces deux énoncés et π_2^1 ; ainsi, le filtrage contextuel ne guide pas davantage le choix à faire. Dans ce cas, nous adoptons l’heuristique qui pénalise les connexions vers des énoncés plus anciens, en faveur des connexions vers des énoncés plus récents : si un énoncé peut être relié à n’importe quel des deux énoncés (mais pas aux deux énoncés simultanément), nous choisissons la connexion vers l’énoncé le plus récemment produit ; ainsi, dans notre cas, nous retenons $ACK(\pi_1^2, \pi_2^1)$. Ainsi, nous observons comment un choix de plus courte portée a été fait, parce que le coût computationnel n’est pris en compte que lorsqu’il s’agit de deux relations rhétoriques de portée égale.

De plus, si nous supposons que la relation de *Consequence* est valide⁹ entre les énoncés π_1^1 et π_1^2 , ensuite, en suivant la même argumentation que ci-dessus, si nous avons accepté les deux structures $ACK(\pi_1^1, \pi_2^1)$ et $ACK(\pi_1^2, \pi_2^1)$, alors nous aurions dû unifier $id(K(3))$ avec $\lambda^- \circ id(K(3))$, ce qui revient à unifier $K(3)$ avec $K(\lambda^-(3))$, ce qui veut dire que nous devrions trouver une entité dont le contenu sémantique coïncide avec le contenu sémantique de la négation de la même entité, ce qui est impossible dans notre cadre formel. Donc, une des deux relations *ACK* doit être supprimée ; ensuite, nous poursuivons la même argumentation que ci-dessus.

Troisièmement, Blanche produit les énoncés π_3^1 et π_3^2 , réalisant tous les deux des actes F^S . Par conséquent, le filtrage pragmatique sort les relations *Elaboration*, *Contrast*, *Consequence* et *Narration* comme possibles entre ces énoncés, ou entre des énoncés antérieurs de Blanche et un de ces énoncés courants (puisque tous les énoncés réalisent le même type d’acte de langage, F^S). De plus, le même filtrage pragmatique sort les relations suivantes comme possibles entre l’énoncé π_2^1 de Léonida et un des énoncés courants de Blanche π_3^1 et π_3^2 : *ACK* et *P-Corr*. Ensuite, le filtrage sémantique détermine les relations suivantes comme valides : *Elaboration*(π_1^2, π_3^1), *Consequence*(π_3^1, π_3^2) et *Narration*(π_3^1, π_3^2). Tout d’abord, il faut vérifier si *Elaboration*(π_1^2, π_3^1) est pertinente au niveau contextuel ; les ensembles de restrictions pour π_1^2 et π_3^1 , tels que déterminés par cette relation rhétorique, sont :

Elaboration(π_1^2, π_3^1) :

$\pi_1^2 : \{id(Elab.1(\epsilon)), id(enounce(1)), \lambda_2^{SubclassOf} \circ id(topic(3))\}$;

$\pi_3^1 : \{id(Elab.2(\epsilon)), id(enounce(2)), \lambda_1^{SubclassOf} \circ id(topic(3))\}$.

Il faut ainsi unifier l’ensemble de restrictions de π_1^2 montré ci-dessus, avec l’ensemble de restrictions pour le même énoncé, tel que déterminé par les relations rhétoriques antérieures $ACK(\pi_1^2, \pi_2^1)$ et *Elaboration*(π_1^1, π_2^1) ; les seuls éléments à paraître problématiques sont $\lambda_2^{SubclassOf} \circ id(topic(3))$ déterminé par *Elaboration*(π_1^2, π_3^1), et $\lambda_1^{SubclassOf} \circ id(topic(3))$, déterminé par *Elaboration*(π_1^1, π_2^1), à cause de l’asymétrie du prédicat *SubclassOf*/2. Mais cela pose des problèmes seulement en apparence, car, à la différence de la situation précédente (où $ACK(\pi_1^1, \pi_2^1)$ avait été supprimée), ici l’argument « 3 » du prédicat *topic*/1 dans les deux éléments dans les ensembles de restrictions de π_1^2 n’a pas une portée globale, parce que les prédicats apparaissent dans des règles *différentes* (qui correspondent à des relations rhétoriques différentes) dans la grammaire abstraite d’attributs associée au discours, et « 3 » n’est pas un atome global. Dans la situation

⁹En fait, le filtrage numérique sort la relation de *Consequence*, car elle a un coût computationnel plus réduit par rapport à *Elaboration*, essentiellement à cause de l’expansion de la fonction *topic*/1 dans *Elaboration*.

précédente avec la relation *ACK*, le problème était que les deux éléments censés être unifiés étaient déterminés par la même relation rhétorique (et ainsi associés à la même règle dans la grammaire du discours), donc « 3 » devait maintenir le même lien (« binding ») pour les deux éléments ; ici cela n'est pas le cas, puisque les éléments résultent des règles différentes dans la grammaire, ce qui équivaut au fait que, ici *topic(3)* se comporte comme une variable locale qui peut être instanciée *différemment* afin que l'unification soit possible. Une autre manière de voir les choses est que dans la situation antérieure avec la relation *ACK*, l'unification des éléments dans les ensembles d'attributs revenait à une inclusion stricte réciproque entre deux topiques d'énoncés, tandis qu'ici la contrainte est qu'il existe deux autres énoncés tels que le topique d'un même (troisième) énoncé soit inclus dans le topique du premier énoncé et inclut le topique du deuxième énoncé, ce qui ne mène pas à une contradiction. Par conséquent, nous pouvons conclure que $Elaboration(\pi_1^2, \pi_3^1)$ passe le filtrage contextuel.

On doit maintenant appliquer le filtrage contextuel sur les relations (validées par le filtrage sémantique) *Narration* et *Consequence* entre les énoncés π_3^1 et π_3^2 . *Narration* détermine les ensembles de restrictions suivants pour ces deux énoncés :

$Narration(\pi_3^1, \pi_3^2)$:

π_3^1 : $\{id(Narration.1(\epsilon)), id(enonce(1)), \lambda_1^{-Disjoint} \circ id(topic(3)), \lambda_1^{smaller} \circ id(\Delta t(4))\}$;

π_3^2 : $\{id(Narration.2(\epsilon)), id(enonce(2)), \lambda_2^{-Disjoint} \circ id(topic(3)), \lambda_2^{smaller} \circ id(\Delta t(4))\}$.

L'ensemble de restrictions de π_3^1 montré ci-dessus devrait s'unifier avec l'ensemble de restrictions du même énoncé, tel que déterminé par la relation $Elaboration(\pi_1^2, \pi_3^1)$; mais $\lambda_1^{SubclassOf} \circ id(topic(3))$ peut être unifié avec $\lambda_1^{-Disjoint} \circ id(topic(3))$, car « 3 » a toujours une portée locale, et ces deux éléments peuvent s'écrire comme $topic(\lambda_1^{SubclassOf}(3))$ et, respectivement, $topic(\lambda_1^{Disjoint}(\lambda^\neg(3)))$ et rien n'interdit deux liages (« bindings ») différents pour « 3 » (on note ces liages – ou liens – différents par 3_1 et 3_2) ainsi que $\lambda_1^{Disjoint}(\lambda^\neg(3_1)) = \lambda_1^{SubclassOf}(3_2)$. Donc, *Narration* est validée entre π_3^1 et π_3^2 .

En ce qui concerne la relation alternative de *Consequence* entre les mêmes deux énoncés, les ensembles de restrictions qu'elle détermine sont :

$Consequence(\pi_3^1, \pi_3^2)$:

π_3^1 : $\{id(Consequence.1(\epsilon)), id(enonce(1)), \lambda^\neg \circ id(K(3))\}$;

π_3^2 : $\{id(Consequence.2(\epsilon)), id(enonce(2)), id(K(3))\}$.

Maintenant, deux choses doivent être vérifiées :

(i) que la relation de *Consequence* peut être validée entre les énoncés π_3^1 et π_3^2 , comme une alternative à la relation de *Narration* entre ces mêmes énoncés. En procédant de la même manière que ci-dessus pour *Narration*, nous obtenons que *Consequence* est aussi valide, en supposant pour l'instant que *Narration* n'est pas, elle aussi, dans la structure rhétorique.

(ii) que la relation de *Consequence* peut être validée entre ces mêmes deux énoncés, en même temps avec *Narration*, c'est-à-dire, que *Consequence* et *Narration* peuvent être validées simultanément ; si cela est vrai, alors le filtrage numérique devra être appliqué, afin de sélectionner une de ces deux relations. Ainsi, nous devons vérifier que les ensembles de restrictions pour π_3^1 et π_3^2 , tels que déterminés par *Narration* et par *Consequence*, peuvent être unifiés. Un cas problématique est, pour π_3^1 , l'unification de $\lambda_1^{-Disjoint} \circ id(topic(3))$ (ou, de façon équivalente, $\lambda^\neg \circ \lambda_1^{Disjoint} \circ id(topic(3))$) et $\lambda^\neg \circ id(K(3))$. Cela revient en fait à unifier $topic(\lambda_1^{Disjoint}(\lambda^\neg(3)))$

avec $K(\lambda^\neg(3))$ mais, du fait que *topic/1* retourne un ensemble, tandis que $K/1$ retourne une forme logique¹⁰, cela est impossible. Donc, les relations de *Narration* et *Consequence* ne peuvent être simultanément valides, mais seulement comme des alternatives exclusives ; il est même inutile de vérifier si les ensembles de restrictions pour π_3^2 , tels que déterminés par ces deux relations, peuvent être unifiés ou pas. Ce « point » dans le discours (les énoncés π_3^1 et π_3^2) est un bon exemple d'un point possible de retour en arrière dans la stratégie de « hiatus coping » si un « hiatus » apparaît plus tard dans le dialogue. Ainsi, entre *Narration* et *Consequence*, nous ne choisissons que la dernière relation, car le coût computationnel de son calcul est plus faible.

Ensuite, Léonida produit le quatrième tour de parole dans ce dialogue ; il contient l'énoncé π_4^1 qui réalise un acte du type F^{FS} (« faire-faire-savoir »). Le filtrage pragmatique sort l'ensemble suivant des relations monologiques potentiellement valides entre π_2^1 et π_4^1 : *Elaboration*, *Background*, *Contrast* et *Consequence* ; le même filtrage pragmatique sort aussi un ensemble de relations dialogiques potentiellement valides entre un ou plusieurs des énoncés π_1^1 , π_1^2 , π_3^1 et π_3^2 , et l'énoncé π_4^1 : *Q-Elab*, *Elab_q*, *Background_q* et *Narration_q*. Cependant, le filtrage sémantique ne sort aucune relation monologique entre les énoncés π_2^1 et π_4^1 ; pour les relations dialogiques, seules *Background_q*(π_1^2, π_4^1) et *Elab_q*($(\pi_1^2; \pi_3^1; \pi_3^2), \pi_4^1$) passent le filtrage sémantique, où $(\pi_1^2; \pi_3^1; \pi_3^2)$ représente le constituant discursif formé avec les énoncés précisées entre les parenthèses, avec les relations rhétoriques qui les relie, ainsi : $(\pi_1^2; \pi_3^1; \pi_3^2) ::= Elaboration(\pi_1^2, \pi_3^1) \wedge Consequence(\pi_3^1, \pi_3^2)$. Nous appliquons maintenant le filtrage contextuel sur ces deux relations rhétoriques. D'abord, *Background_q*(π_1^2, π_4^1) détermine les ensembles de restrictions suivants pour les deux énoncés¹¹ :

Background_q(π_1^2, π_4^1) :

$\pi_1^2 : \{id(Background_q.1(\epsilon)), id(enonce(1)), id(entails.2(6))\}$;

$\pi_4^1 : \{id(Background_q.2(\epsilon)), id(question(2)), \lambda_2^{setof} \circ id(answer.2(3))\}$.

Cela est vrai parce que la sémantique de *Background_q*(π_1^2, π_4^1) a été exprimée sous la forme du programme logique suivant :

Background_q(π_1^2, π_4^1) : $-enonce(\pi_1^2), question(\pi_4^1), setof(\chi, answer(\chi, \pi_4^1), Set),$
 $memberOf(_ \chi, Set), enonce(_ \chi), entails(_ \chi, \pi_1^2)$.

Comme il résulte de l'ensemble de restrictions de π_1^2 , cet ensemble peut être unifié avec des ensembles de restrictions antérieurs de cet énoncé, tels que déterminés par *Elaboration*(π_1^1, π_1^2), *ACK*(π_1^2, π_2^1) et *Elaboration*(π_1^2, π_3^1). Nous pouvons ainsi conclure que *Background_q* est valide entre les énoncés π_1^2 et π_4^1 .

Ensuite, nous étudions si *Elab_q*($(\pi_1^2; \pi_3^1; \pi_3^2), \pi_4^1$) est aussi valide. Pour ce faire, nous devons montrer comment les relations rhétoriques *imbriquées* sont manipulées : en fait elles sont manipulées comme n'importe quel autre prédicat imbriqué (considéré comme un fait vrai si une telle relation rhétorique avait déjà passé les trois filtres – pragmatique, sémantique et contextuel), *via* les fonctions sélecteurs qui leur sont appliquées. Les seules différences sont que, dans notre cas, la relation rhétorique *Elab_q* détermine quatre ensembles de restrictions, pour chaque variable globale (c'est-à-dire, chaque énoncé), et que l'opérateur de conjonction qui « relie »

¹⁰En fait, la différence entre ces deux fonctions est que K retourne des foncteurs, connecteurs logiques, atomes et un relation d'ordre entre eux, tandis que *topic* ne retourne que des atomes.

¹¹En appliquant la sémantique de *Background_q*, nous avons éclaté la sémantique de la relation *Background* (qui apparaît comme constituant de *Background_q*) en termes des prédicats discursifs qui la composent.

les « points » de discours dans un constituant discursif est distribué sur les prédicats où ce constituant apparaît comme argument. Par exemple, s'il y a un prédicat comme $p(\Pi)$, où Π est un constituant discursif de la forme $\Pi ::= \rho_i(\pi_j, \pi_k) \wedge \dots \wedge \rho_l(\pi_m, \pi_n)$, alors $p(\Pi)$ peut s'écrire comme : $p(\Pi) = p(\rho_i(\pi_j, \pi_k)) \wedge \dots \wedge p(\rho_l(\pi_m, \pi_n))$. Par conséquent, $Elab_q((\pi_1^2; \pi_3^1; \pi_3^2), \pi_4^1)$ peut s'écrire comme le programme logique suivant :

$$Elab_q((\pi_1^2; \pi_3^1; \pi_3^2), \pi_4^1) : \text{--enounce}(Elaboration(\pi_1^2, \pi_3^1)), \text{enounce}(Consequence(\pi_3^2, \pi_3^1)), \\ \text{question}(\pi_4^1), \text{setof}(\chi, \text{answer}(\chi, \pi_4^1), Set), \text{memberOf}(_ \chi, Set), \\ \text{enounce}(_ \chi), (\text{SubclassOf}(\text{topic}(_ \chi), [\text{topic}(Elaboration(\pi_1^2, \pi_3^1)), \\ \text{topic}(Consequence(\pi_3^2, \pi_3^1))]); \text{equals}(\text{topic}(_ \chi), \\ [\text{topic}(Elaboration(\pi_1^2, \pi_3^1)), \text{topic}(Consequence(\pi_3^2, \pi_3^1))])).$$

Dans ce programme, nous avons mis dans une liste (à la Prolog) le résultat de la fonction `topic/1` appliquée sur la conjonction des relations rhétoriques (dans un constituant discursif), parce qu'en fait cette fonction retourne une liste, donc l'effet de distribuer l'opérateur de conjonction sur `topic/1` est justement la concaténation, dans une liste, des résultats que cette fonction retourne sur chaque membre de la conjonction. Cependant, dans cette situation une question apparaît : comment les fonctions sélecteurs opèrent-ils sur les listes ? La réponse est que la liste est traitée tout simplement comme un argument multidimensionnel, dont les composants sont sélectionnés comme les arguments d'un foncteur multinaire. Les ensembles d'attributs pour les énoncés en question sont indiqués ci-dessous :

$$Elab_q((\pi_1^2; \pi_3^1; \pi_3^2), \pi_4^1) : \\ \pi_1^2 : \{\lambda_1^{Elab_q} \circ id(Elaboration.1(\epsilon)), \lambda^{enounce} \circ id(Elaboration.1(1)), \lambda_2^{SubclassOf}.1 \circ \lambda^{topic} \circ \\ id(Elaboration.1(7)), \lambda_2^{equals}.1 \circ \lambda^{topic} \circ id(Elaboration.1(8))\}; \\ \pi_1^3 : \{\lambda_1^{Elab_q} \circ id(Elaboration.2(\epsilon)), \lambda_1^{Elab_q} \circ id(Consequence.1(2)), \lambda^{enounce} \circ id(Elaboration. \\ 2(1)), \lambda^{enounce} \circ id(Consequence.1(2)), \lambda_2^{SubclassOf}.1 \circ \lambda^{topic} \circ id(Elaboration.2(7)), \\ \lambda_2^{SubclassOf}.2 \circ \lambda^{topic} \circ id(Consequence.1(7)), \lambda_2^{equals}.1 \circ \lambda^{topic} \circ id(Elaboration.2(8)), \\ \lambda_2^{equals}.2 \circ \lambda^{topic} \circ id(Consequence.1(8))\}; \\ \pi_3^2 : \{\lambda_1^{Elab_q} \circ id(Consequence.2(\epsilon)), \lambda^{enounce} \circ id(Consequence.2(2)), \lambda_2^{SubclassOf}.2 \circ \lambda^{topic} \circ \\ id(Consequence.2(7)), \lambda_2^{equals}.2 \circ \lambda^{topic} \circ id(Consequence.2(8))\}; \\ \pi_4^1 : \{id(Elab_q.2(\epsilon)), id(question(3)), \lambda_2^{setof} \circ id(answer.2(4))\}.$$

Maintenant, s'il y a au moins deux ensembles de restrictions pour au moins une des trois premières variables globales qui ne peuvent pas être unifiés, alors la relation $Elab_q$ peut être supprimée. Par exemple, nous essayons d'unifier deux ensembles de restrictions pour π_1^2 ; pour cela, nous essayons d'unifier $\lambda_2^{SubclassOf}.1 \circ \lambda^{topic} \circ id(Elaboration.1(7))$ dans l'ensemble de restrictions montré ci-dessus, avec $\lambda_2^{SubclassOf} \circ id(topic(3))$, dans l'ensemble de restrictions déterminé par $Elaboration(\pi_1^2, \pi_3^1)$. Cela revient à résoudre l'équation suivante :

$$Elaboration.1(\lambda^{topic}(\lambda_2^{SubclassOf}.1(7))) = \text{topic}(\lambda_2^{SubclassOf}(3)).$$

Même si nous pouvons trouver une instantiation pour « 3 » et « 7 » ainsi que $\lambda_2^{SubclassOf}.1(7)$ s'unifie avec $\lambda_2^{SubclassOf}(3)$, le problème est que `topic/1` retourne un ensemble, tandis que $Elaboration$ revient à une forme logique qui, au meilleur cas, inclut de manière stricte `topic/1` (il n'y a pas de sémantique d'une relation rhétorique qui soit composée seulement d'une conjonction de fonctions `topic/1`). Conséquemment, ces deux termes ne peuvent être unifiés dans notre cadre ; donc, $Elab_q((\pi_1^2; \pi_3^1; \pi_3^2), \pi_4^1)$ est supprimée.

En conclusion, la structure rhétorique qui reste valide, après tous les filtrages, consiste dans $Elaboration(\pi_1^1, \pi_1^2)$ ou $Consequence(\pi_1^1, \pi_1^2)$ (si le filtrage numérique est appliqué, alors seule la dernière de ces relations sera retenue), $ACK(\pi_1^2, \pi_2^1)$, $Elaboration(\pi_2^2, \pi_3^1)$, $Consequence(\pi_3^1, \pi_3^2)$ ou $Narration(\pi_3^1, \pi_3^2)$ (si le filtrage numérique est appliqué, alors seule la première de ces relations sera retenue), et $Background_q(\pi_1^2, \pi_4^1)$.

4.4 Conclusions

Dans ce chapitre nous avons proposé un filtrage contextuel des relations de discours en structuration rhétorique pour la génération des répliques en dialogue. Ce processus prend en compte la structure rhétorique de tout le discours existant à un moment donné, pour ainsi la mettre à jour avec le contenu sémantique d'un nouvel énoncé. Le critère de filtrage réside dans l'unifiabilité de tous les ensembles de restrictions pour les énoncés, qui jouent le rôle de variables globales dans la grammaire abstraite d'attributs de tout le discours. Un autre filtrage, optionnel et d'importance plutôt combinatoire, a été également proposé pour évaluer de manière relative les relations rhétoriques, selon le coût computationnel requis pour les calculer. Ce coût est vu comme une incidence directe de la complexité de la grammaire abstraite d'attributs qui correspond à chaque relation rhétorique.

Le filtrage contextuel présenté dans ce chapitre peut être considéré comme une alternative au principe de Maximisation de la cohérence discursive (MDC – « Maximize Discourse Coherence ») dans la SDRT d'origine, mais au lieu de calculer la structure de discours la plus cohérente, nous calculons la structure la plus « simple », encore cohérente¹². Le mot « simple » veut dire ici qu'un effort computationnel minimal devrait être fait pour calculer cette structure rhétorique.

En conclusion, notre approche en structuration rhétorique pour la génération des répliques en dialogue homme-machine formalise les processus de mise à jour de la structure rhétorique, selon les énoncés produits par les utilisateurs, ou *censés être* produits par la machine, d'une manière séquentielle : tout d'abord, pour une paire d'énoncés donnée, nous retenons un nombre relativement faible de relations rhétoriques *potentiellement valides* entre ces énoncés ; ensuite, pour la même paire d'énoncés, nous vérifions, *via* un processus d'inférence basé sur un ensemble de prédicats discursifs indépendants de la tâche et d'une ontologie de tâches, la validité sémantique de l'ensemble des relations rhétoriques potentiellement valides, pour ainsi en retenir un ensemble encore plus restreint de relations dites *potentiellement pertinentes* ; ensuite, pour la même paire d'énoncés et pour tous les énoncés antérieurs avec les relations rhétoriques qui les relient entre eux, nous vérifions, *via* une approche qui consiste à convertir les sémantiques des relations rhétoriques dans des grammaires abstraites d'attributs, les relations rhétoriques sorties comme potentiellement pertinentes par le filtrage sémantique, pour ainsi retenir un ensemble encore plus restreint (qui contient, nous l'espérons, au plus un élément) de relations rhétoriques *pertinentes*. Enfin, si, étant donnée une paire d'énoncés, l'ensemble des relations rhétoriques pertinentes a strictement plus d'un élément, alors le filtrage numérique est appliqué, afin de ne retenir qu'une seule (au plus) relation rhétorique, celle dont le coût de calcul est le plus réduit.

¹²Cette cohérence est garantie par les filtrages sémantique et contextuel.

Chapitre 5

Structuration du discours en génération pour le dialogue multi-locuteurs

5.1 Introduction

Même si les dialogues entre un ordinateur et un *seul* partenaire humain sont étudiés de manière tant théorique que pratique dans un contexte assez mûr [116], la situation où l'ordinateur est censé interagir avec plusieurs interlocuteurs humains à la fois, est encore peu étudiée.

Pour aller au delà du dialogue homme-machine classique, entre deux partenaires, il existe essentiellement deux possibilités :

- le dialogue **multi-session**, où la machine se trouve impliquée dans des dialogues parallèles avec plusieurs locuteurs humains ; ces dialogues sont indépendants au sens où les locuteurs n'interagissent pas entre eux et n'ont pas accès aux dialogues entre la machine et les autres partenaires de dialogue. Ce type d'interaction est particulièrement intéressant pour des situations supposant l'accès concurrent à un ensemble limité de ressources (par exemple, la réservation des salles dans une entreprise) ; par conséquent, dans ce cas nous avons affaire à plusieurs dialogues *classiques* à deux partenaires alternés, sur lesquels l'ordinateur devrait maintenir une représentation cohérente. Même si cela ne représente pas un véritable dialogue à plusieurs locuteurs, il y a relativement peu de recherches, au niveau mondial, sur ce sujet. Par exemple, l'état de l'art actuel est représenté par le système PVE (« Portail Vocal pour l'Entreprise ») [122]. Dans ce système on gère de multiples sessions d'interaction, au niveau du contrôle du dialogue, dans une approche inspirée de la théorie de jeux, où les tours dialogiques de la machine sont évalués *via* des gains (ou utilités) et dépendent, en même temps, du contexte de la tâche (volume des ressources, rôles des interlocuteurs, etc.) et des actes de langage réalisés par les locuteurs.
- le dialogue **multi-locuteurs** ou à plusieurs locuteurs, où la machine est *simultanément* impliquée dans des dialogues avec plusieurs locuteurs ; à l'instar du dialogue multi-session, la machine doit garder une vue cohérente sur les dialogues ; pourtant, il y a une différence

majeure par rapport aux conversations multi-session : en interaction multi-locuteurs, les dialogues sont simultanés, tous les locuteurs étant au même endroit et ayant accès à *tous* les énoncés des interlocuteurs. C'est pour ces raisons que la modélisation de ce type d'interaction est particulièrement difficile. Cependant, à partir des années 2000 environ, il y a de plus en plus de recherches dans ce sens, recherches qui essaient d'étudier la portabilité des modèles conçus pour le dialogue entre deux partenaires, au dialogue multi-locuteurs [64], ou d'analyser des corpus de dialogues à plusieurs locuteurs afin de déterminer les différences entre les dialogues à deux et les dialogues multi-locuteurs [142], ou même de modéliser formellement certains aspects du dialogue à plusieurs locuteurs (comme le contrôle du dialogue), et concernant seulement quelques aspects (comme le contexte *partagé* entre les interlocuteurs) [98].

Ce chapitre abordera plusieurs problèmes sur la modélisation des aspects sémantiques et pragmatiques concernant la structuration rhétorique pour la génération des énoncés en dialogues multi-locuteurs orientés service. Ainsi, nous proposons tout d'abord un cadre pour représenter à un niveau rhétorique les dialogues à plusieurs locuteurs ; ensuite, plusieurs situations particulières, pertinentes pour les applications du type service public, sont présentées de manière détaillée. Pour ces situations, la représentation formelle est complétée avec des traitements procéduraux, instanciés dans des algorithmes qui pilotent le calcul des structures rhétoriques qui spécifient les dialogues. Ces structures rhétoriques seront ensuite utilisées pour contraindre le choix des formes linguistiques appropriées, pour exprimer une certaine *intention communicative*. Ainsi, ce chapitre étend au dialogue multi-locuteurs les travaux présentés dans les chapitres 3 et 4 de cette thèse, en proposant un cadre pour la mise à jour des structures rhétoriques, lorsqu'il s'agit, pour la machine, de générer un tour de parole dans un contexte de dialogue avec plusieurs interlocuteurs.

Ce chapitre est structuré de la manière suivante : tout d'abord, nous définissons dans le paragraphe suivant un cadre général pour représenter les dialogues à plusieurs locuteurs d'un point de vue rhétorique ; ensuite, nous allons focaliser notre attention sur deux cas particulièrement importants, selon l'« implication » du locuteur en conversation. L'analyse sera ensuite limitée à la perspective que la machine se construit sur la situation de communication ; plus précisément, seul le cas où la machine adopte la perspective « impliquée » sera discuté de manière plus détaillée. Ce modèle (limité) est ensuite spécifié à un niveau procédural, *via* des algorithmes pour la mise à jour des structures rhétoriques, lorsqu'il s'agit, pour la machine, de générer un tour de parole. Enfin, la spécification procédurale sera illustrée *via* des exemples typiques de dialogues multi-locuteurs.

5.2 Cadre général

5.2.1 Conventions

Nous définissons tout d'abord un ensemble de notations, que nous allons utiliser tout au long de ce chapitre :

- $L_1, \dots, L_N, L_\alpha, L_\beta, \dots ::=$ locuteur humain dont l'identité est i , $i \in \{1, \dots, N\}$ ou α, β , etc. ;
- $M ::=$ la machine ;

- $v_{ij}, v_{\alpha\beta} ::=$ chaîne de communication entre les locuteurs L_i et L_j (c'est-à-dire, si un dialogue a lieu entre ces deux locuteurs, alors cela se passe *via* la chaîne v_{ij}); $v_{ij} \equiv v_{ji}$ pour tous i et j ;
- $SDRS_{\alpha}^{\beta\gamma} ::=$ structure rhétorique (SDRS) *perçue* par le locuteur L_{α} , pour le dialogue qui a lieu *via* la chaîne $v_{\beta\gamma}$;
- $SDRS_{ijk} ::=$ réunion de toutes les SDRS perçues par le locuteur L_k sur les chaînes v_{ij} ;
- $SDRS_{ij} ::= \bigcup_{\forall k} SDRS_{ijk}$;
- $\pi(\alpha, \beta) ::=$ étiquette d'un énoncé produit par L_{α} et destiné à L_{β} ;
- $\pi(\alpha, B) ::=$ étiquette d'un énoncé produit par L_{α} et destiné à l'ensemble de locuteurs L_{β} : $\beta \in B$;
- $t(\pi) ::=$ indice ordinal de l'énoncé libellé π , dans la suite d'énoncés produits par l'émetteur de π , dans le dialogue où ce locuteur participe couramment;
- $\text{emitter}(\pi) ::=$ l'émetteur (c'est-à-dire, celui qui le produit) de l'énoncé libellé π ;
- $\text{equals}(n, m) ::=$ prédicat vrai si et seulement si $n = m$ pour des valeurs numériques, ou $n \equiv m$, pour d'autres types d'atomes;
- $\text{turn}(\pi, \alpha) ::= \text{equals}(\text{emitter}(\pi), L_{\alpha}) \wedge t(\pi)$;
- $K(\pi) ::=$ formule logique qui exprime la sémantique de l'énoncé libellé π ;
- $SDRS^{\alpha}(i) ::=$ sous-structure rhétorique qui correspond à la contribution du locuteur L_{α} au i ème tour de parole;
- $\sigma(\alpha, \beta) ::= \{(\alpha, \beta); (\beta, \alpha)\}$, l'ensemble des permutations du doublet (α, β) ;
- $MP ::=$ « relation » rhétorique qui relie le dernier énoncé d'un locuteur humain, au premier énoncé du locuteur humain suivant (donc, à l'exception de la machine) dans un tour de parole; il faut noter que cette relation ne sert qu'à signaler une situation de conversation à plusieurs, où deux interlocuteurs humains sont en train d'(inter)agir en séquence;
- $\rho_m^{(c)} ::=$ relation rhétorique monologique de confirmation (voir chapitre 3);
- $\rho_m^{(-c)} ::=$ relation rhétorique monologique de contradiction;
- $\rho_d^{(c)} ::=$ relation rhétorique dialogique de confirmation;
- $\rho_d^{(-c)} ::=$ relation rhétorique dialogique de contradiction;
- les symboles logiques $\exists, \forall, \Rightarrow, \neg, \wedge, \vee$ gardent leur sens usuels;
- les symboles mathématiques $<, \leq, =, \geq, >, \cup, \cap, \emptyset, \subseteq, \subset, \supseteq, \in, \ni, \setminus$ gardent également leur sens usuel.

Ces notations nous permettront de développer le cadre formel concernant la *perspective rhétorique* sur les dialogues à plusieurs locuteurs. Nous allons faire cela dans le paragraphe suivant.

5.2.2 Situations de dialogue à plusieurs locuteurs

Pour les traitements formels présentés dans ce paragraphe, nous supposons qu'il y a N locuteurs engagés dans une conversations à plusieurs; cela est illustré dans la figure 5.1, où nous avons représenté les chaînes de communication potentiellement établies entre les paires de locuteurs.

Toutes les structures rhétoriques correspondantes à des dialogues entre des paires de locuteurs sont précisées ci-dessous :

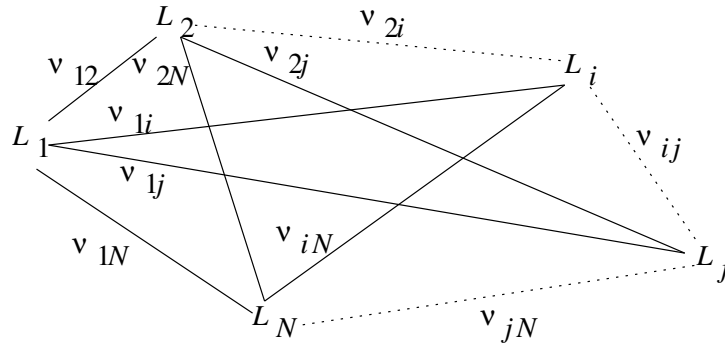


FIG. 5.1 – Situation de dialogue à plusieurs locuteurs

$$SDRS_{1N} = \bigcup_{i=1}^N \left(\bigcup_{\substack{j=1 \\ j \neq i}}^N \left(SDRS_i^{ij} \cup \left(\bigcup_{\substack{k=1 \\ k \neq j, i}}^N SDRS_i^{jk} \right) \right) \right)$$

Par exemple, pour trois locuteurs L_i , L_j et L_k , l'équation montrée ci-dessus a la forme particulière suivante :

$$SDRS_{ik} = SDRS_i^{ij} \cup SDRS_i^{ik} \cup SDRS_j^{ji} \cup SDRS_j^{jk} \cup SDRS_k^{ki} \cup SDRS_k^{kj} \cup SDRS_i^{jk} \cup SDRS_j^{ik} \cup SDRS_k^{ij}$$

Le cadre formel proposé ici est basé sur deux postulats :

Postulat 1. (*Perspective rhétorique commune*) Deux locuteurs impliqués dans un dialogue se construisent la même structure rhétorique pour celui-ci. Formellement,
 $\forall L_i, L_j : \text{equals}(SDRS_i^{ij}, SDRS_j^{ij})$

Postulat 2. (*Réflexivité du dialogue*) Pour trois locuteurs, dont deux impliqués dans un dialogue et le troisième observant les deux premiers, la représentation rhétorique calculée par ce troisième locuteur est invariante par rapport à l'ordre dans laquelle il considère les deux premiers locuteurs. Formellement,

$$\forall L_i, L_j, L_k : \text{equals}(SDRS_k^{ij}, SDRS_k^{ji})$$

Ces deux postulats disent essentiellement que (i) deux locuteurs en dialogue interprètent de la même manière ce dialogue, au sens où ils se construisent deux structures rhétoriques logiquement équivalentes, et (ii) pour un observateur extérieur, un dialogue entre deux locuteurs, L_A et L_B est équivalent au même dialogue entre les locuteurs L_B et L_A , pour le même intervalle de temps.

Evidemment, le deuxième postulat ne représente qu'un détail technique, permettant de permuter les indices supérieurs dans les symboles utilisés pour représenter les SDRS dialogiques, tandis que le premier postulat est réellement intéressant, dans la mesure où il faut préciser les conditions dans lesquelles deux locuteurs se forment la même représentation rhétorique sur un dialogue qui les engage tous les deux. Comme par ailleurs Asher le mentionne lui-même dans [12], il n'est pas évident comment on peut trouver un ensemble de conditions générales, nécessaires et suffisantes pour qu'un couple de locuteurs se forment la même représentation discurs-

sive du dialogue entre eux. Il n'en reste pas moins que, lorsque nous substituons l'ambition de construire une théorie « complète » de la performance rhétorique des individus humains, par un but plus modeste, de mettre en place un cadre formel implémentable dans des systèmes de dialogue finalisés par la tâche, nous pouvons faire quelques approximations.

Ainsi, nous pouvons tout d'abord supposer que les compétences inférentielles des deux interlocuteurs sont égales (ce qui serait particulièrement pertinent pour un dialogue entre des agents artificiels, où le même moteur d'inférence est implémenté dans chacun de ces interlocuteurs). Cette approximation pourrait être correcte aussi pour des interlocuteurs humains, si ces derniers partagent beaucoup de traits communs (comme par exemple, l'arrière-plan, le métier, la formation, etc.), ou si, de plus, ils limitent leur dialogue à un sujet ou domaine bien restreint, où chacun des interlocuteurs opère avec un ensemble explicite de règles et de principes (voir dans ce sens les dialogues spécialisés entre des sapeurs pompiers, par exemple).

Ensuite, une fois admise cette première approximation sur la symétrie des capacités inférentielles des interlocuteurs, leur capacité de calculer une structure rhétorique correcte repose sur l'étendue de leurs connaissances, autrement dit, dans le contexte d'un dialogue (personne-système) finalisé par la tâche, sur l'étendue de l'ontologie de tâches. La relation entre les prédicats et les règles dans l'ontologie de tâches d'une part, et l'ensemble des relations rhétoriques calculables entre des énoncés en dialogue d'autre part, n'est pas évidente, mais, en admettant que le calcul de la structure rhétorique en dialogue bi-locuteurs se fait selon les principes décrits dans les chapitres 3 et 4 de cette thèse, le calcul d'une relation rhétorique entre une paire d'énoncés revient à « comprendre » ces énoncés. La compréhension des énoncés peut être quantifiée, nous supposons, en relation avec l'ontologie de tâches, au niveau des relations rhétoriques du type *Background* qui s'établissent entre les propositions dans l'ontologie de tâches et les énoncés en dialogue. Ainsi, en supposant un dialogue qui contient un ensemble d'énoncés libellés π_1, \dots, π_n , et un ensemble de propositions dans l'ontologie de tâches d'un locuteur L_A , libellées $\tau_1^{(A)}, \dots, \tau_p^{(A)}$, aussi qu'un ensemble de propositions dans l'ontologie de tâches d'un locuteur L_B , libellées $\tau_1^{(B)}, \dots, \tau_q^{(B)}$, il s'agit d'étudier les relations du type *Background*($\pi_j, \tau_i^{(A)}$) que L_A calcule, et les relations *Background*($\pi_j, \tau_k^{(B)}$), que L_B calcule. Autrement dit, il faut étudier les propositions dans les ontologies de tâches, qui constituent des arrière-plans aux énoncés dans le dialogue. Bien évidemment, ce sujet requiert des études extensives, mais nous conjecturons pour l'instant que l'équivalence des ensembles de relations de *Background* entre les propositions dans l'ontologie de tâches et les énoncés dans le dialogue, pour une paire de locuteurs, constitue une prémisse suffisante (quoique pas forcément nécessaire – il faudra étudier cela) pour que les deux interlocuteurs calculent la même SDRS sur le dialogue formé des énoncés π_1 à π_n . Formellement, cela peut être décrit comme une règle du type (Σ_ρ représente la sémantique formelle de la relation rhétorique ρ , selon les définitions précisées dans le chapitre 3 de cette thèse) :

$$(\forall \pi_i \in \text{Dialogue entre } L_A \text{ et } L_B, i \in \{1, \dots, n\} \exists \tau_j^{(A)}, j \in \{1, \dots, p\} : \Sigma_{\text{Background}(\pi_i, \tau_j^{(A)})} = \text{TRUE} \wedge \exists \tau_k^{(B)}, k \in \{1, \dots, q\} : \Sigma_{\text{Background}(\pi_i, \tau_k^{(B)})} = \text{TRUE} \wedge \text{equals}(K(\tau_j^{(A)}), K(\tau_k^{(B)}))) \Rightarrow \text{equals}(S\text{DRS}_A^{AB}, S\text{DRS}_B^{AB}).$$

De toute manière, le sujet de l'équivalence des représentation discursives de deux locuteurs en dialogue reste ouvert à des recherches futures, car il faudra établir et valider les conditions nécessaires et suffisantes pour que cette équivalence ait lieu.

En admettant les deux postulats précisés ci-dessus, l'ensemble des SDRS qui représentent la structure rhétorique de la situation de dialogue à plusieurs locuteurs devient :

$$SDRS_{1N} = \bigcup_{i=1}^N \left(\bigcup_{j=i+1}^N \left(SDRS_i^{ij} \cup \left(\bigcup_{\substack{k=1 \\ k \neq j, i}}^N SDRS_i^{jk} \right) \right) \right)$$

Pour le cas particulier où trois locuteurs sont considérés, l'expression se simplifie davantage :
 $SDRS_{ik} = SDRS_i^{ij} \cup SDRS_i^{ik} \cup SDRS_j^{jk} \cup SDRS_i^{jk} \cup SDRS_j^{ik} \cup SDRS_k^{ij}$.

En ce qui concerne les perspectives que les locuteurs peuvent adopter sur le dialogue dans lequel ils participent, nous pouvons mettre en évidence deux cas :

I. Perspective « impliquée » : Le locuteur participe au dialogue dont il se fait une représentation discursive (sous forme d'une SDRS). Nous nous intéressons aux structures rhétoriques du type $SDRS_\alpha^{\alpha\beta}$, pour le locuteur L_α qui (se) construit une représentation du dialogue qu'il a avec le locuteur L_β . Ainsi, pour deux couples de locuteurs (L_α, L_β) et (L_φ, L_ψ) , il y a deux sous-cas :

1. $SDRS_\alpha^{\alpha\beta} \cap SDRS_\varphi^{\varphi\psi} = \emptyset$. Dans ce cas plusieurs dialogues se déroulent en parallèle, ce qui est réductible au dialogue traditionnel (entre deux locuteurs) ; ce cas n'est pas intéressant ici ;
2. $SDRS_\alpha^{\alpha\beta} \cap SDRS_\varphi^{\varphi\psi} \neq \emptyset$. Dans ce cas nous avons affaire à un véritable dialogue multi-locuteurs, qui va être analysé de manière détaillée dans ce chapitre.

Du point de vue d'un locuteur impliqué dans un dialogue à plusieurs locuteurs (cas I.2. mentionné ci-dessus), si nous désignons par ρ l'étiquette d'une relation rhétorique, alors nous pouvons écrire de manière explicite la SDRS que L_α se construit pour le dialogue qu'il entretient avec un autre locuteur, L_β :

$$SDRS_\alpha^{\alpha\beta} = \left(\bigcup_t \text{tour de parole } (\pi(\sigma(\alpha, \beta)) : \text{equals}(t(\pi), t)) \right) \cup \left(\bigcup_{t < t'} \text{tours de parole } (\rho(\pi(\sigma(\alpha, \beta)), \pi'(\sigma(\alpha, \beta))) : \text{equals}(t(\pi), t) \wedge \text{equals}(t(\pi'), t')) \right)$$

Cette dernière formule veut dire, de manière informelle, qu'une SDRS construite par le locuteur L_α sur un dialogue qu'il entretient avec le locuteur L_β , consiste, tout simplement, dans une réunion de tous les énoncés dans tous les tours de parole dans le dialogue, tels qu'il existe des relations rhétoriques entre eux (ainsi que le graphe ayant comme nœuds les énoncés et comme arêtes les relations rhétoriques soit connexe – cf. la définition d'une SDRS que nous adoptons dès les chapitres 3 et 4 de cette thèse), et des relations rhétoriques qui existent entre les paires d'énoncés (ou de constituants discursifs¹).

Ainsi, une intersection non-vide de deux structures rhétoriques revient à l'existence d'énoncés qu'un locuteur destine à plusieurs locuteurs ; il faut noter que l'absence de sous-graphes communs à deux SDRS n'entraîne pas le fait que ces structures soient disjointes, selon notre définition :

¹Un constituant discursif se définit de manière récursive à partir de paires d'énoncés connectés par des relations rhétoriques : deux énoncés reliés par une relation rhétorique forment un constituant discursif ; un troisième énoncé relié *via* une relation rhétorique au premier constituant discursif, forme avec ce dernier un nouveau constituant discursif.

$$\neg \text{equals}(S DRS_{\alpha}^{\alpha\beta} \cap S DRS_{\varphi}^{\varphi\psi}, \emptyset) \Leftrightarrow \left(\begin{array}{c} \alpha - \varphi \\ \alpha - \beta \end{array} \right) \vee \left(\begin{array}{c} \alpha - \psi \\ \alpha - \beta \end{array} \right) \vee \left(\begin{array}{c} \beta - \varphi \\ \beta - \alpha \end{array} \right) \vee \left(\begin{array}{c} \beta - \psi \\ \beta - \alpha \end{array} \right) \vee \left(\begin{array}{c} \varphi - \alpha \\ \varphi - \psi \end{array} \right) \vee \left(\begin{array}{c} \varphi - \beta \\ \varphi - \psi \end{array} \right) \vee \left(\begin{array}{c} \psi - \alpha \\ \psi - \varphi \end{array} \right) \vee \left(\begin{array}{c} \psi - \beta \\ \psi - \varphi \end{array} \right).$$

Dans la dernière notation condensée, pour trois locuteurs L_X, L_Y, L_Z , les expressions parenthiques ont la signification suivante :

$$\left(\begin{array}{c} X - Y \\ X - Z \end{array} \right) ::= \exists \pi, \pi' : \pi(X, Y) \wedge \pi'(X, Z) \wedge \text{equals}(\text{turn}(\pi, X), \text{turn}(\pi', X)) \wedge \text{equals}(K(\pi), K(\pi')).$$

De manière informelle, l'expression parenthique à gauche de la dernière expression signifie qu'il existe un énoncé produit par un locuteur (L_X), énoncé qui est destiné à deux locuteurs différents (L_Y et L_Z).²

II. Perspective « extérieure » : Le locuteur ne participe pas au dialogue dont il se construit une représentation discursive, il ne fait qu'*observer* un dialogue entre deux autres locuteurs. Même si le locuteur ne participe pas au dialogue, il se fait une représentation (rhétorique) de celui-ci, parce qu'il doit être capable de s'y impliquer à tout moment, en produisant des répliques à bon escient. Dans ce cas nous nous intéressons aux structures rhétoriques du type $S DRS_{\alpha}^{\beta\gamma}$. Ainsi, pour deux triples de locuteurs ($L_{\alpha}, L_{\beta}, L_{\gamma}$) et ($L_{\eta}, L_{\varphi}, L_{\psi}$) il y a deux sous-cas :

1. $S DRS_{\alpha}^{\beta\gamma} \cap S DRS_{\eta}^{\varphi\psi} = \emptyset$. Ce cas est réductible à des dialogues simultanés indépendants qui ont lieu en parallèle, ou à des dialogues multi-session (si, par exemple, $\text{equals}(L_{\beta}, L_{\varphi})$) ; par conséquent, ce cas ne sera pas détaillé dans ce chapitre ;
2. $S DRS_{\alpha}^{\beta\gamma} \cap S DRS_{\eta}^{\varphi\psi} \neq \emptyset$. Ce cas représente une véritable situation de dialogue multi-locuteurs, où deux locuteurs partagent leurs perspectives (comme des observateurs) de deux autres dialogues, impliquant d'autres paires de locuteurs. Par conséquent, ce cas sera analysé dans ce chapitre.

En ce qui concerne les représentations rhétoriques calculées par les locuteurs qui observent une paire de dialogues multi-locuteurs (le cas II.2 ci-dessus), nous pouvons détailler encore deux sous-cas :

1. $\text{equals}((L_{\beta}, L_{\gamma}), (L_{\varphi}, L_{\psi})) \Rightarrow S DRS_{\alpha}^{\beta\gamma} \cap S DRS_{\eta}^{\beta\gamma} \neq \emptyset$. Cela est normal, puisqu'il s'agit de deux perspectives différentes sur le même dialogue (entre L_{β} et L_{γ}) ; il y a encore trois sous-cas :
 - (a) les deux « observateurs » L_{α} et L_{η} partagent la perspective « impliquée » du locuteur L_{β} sur le dialogue qui a lieu *via* la chaîne de communication $\nu_{\beta\gamma}$:
$$S DRS_{\alpha}^{\beta\gamma} \equiv S DRS_{\eta}^{\beta\gamma} \equiv S DRS_{\beta}^{\beta\gamma}.$$
Cela peut arriver si L_{α} et L_{η} ont en commun un arrière-plan riche avec les locuteurs L_{β} et L_{γ} , tout en étant capables d'accéder au dialogue que ces derniers locuteurs maintiennent ;

²En fait, la signification de la forme logique exprimée ci-dessus est qu'il existe deux énoncés libellés π et π' tels que π est produit par L_X et destiné à L_Y , et π' est émis par le locuteur L_X et destiné à L_Z , ainsi que π et π' sont produits au même moment par L_X , et leurs sémantiques sont équivalentes.

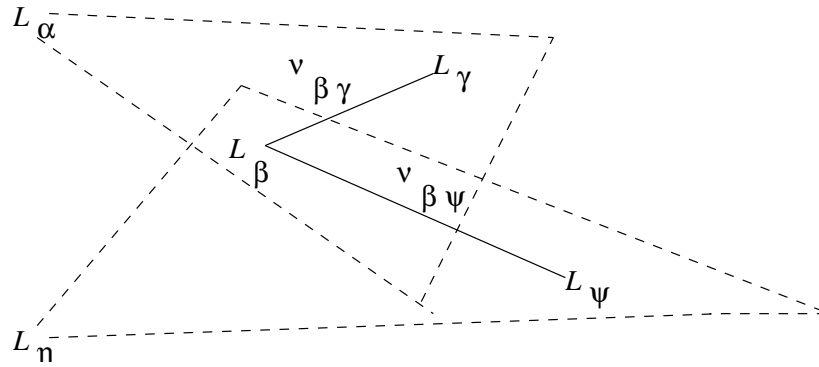


FIG. 5.2 – Vue partagée sur les tours de parole adressés simultanément à deux locuteurs

- (b) les deux « observateurs » L_α et L_η se partagent une perspective partielle, incluse dans la perspective impliquée de L_β , sur le dialogue qui a lieu *via* la chaîne $v_{\beta\gamma}$:

$$S DRS_\alpha^{\beta\gamma} \cap S DRS_\eta^{\beta\gamma} \subset S DRS_\beta^{\beta\gamma} ;$$

- (c) les deux « observateurs » L_α et L_η se partagent une perspective sur le dialogue entre L_β et L_γ qui n'a rien en commun avec la perspective que L_β et L_γ se *partagent* sur ce dialogue (en vertu du postulat 1, de la perspective rhétorique commune) :

$$S DRS_\alpha^{\beta\gamma} \cap S DRS_\eta^{\beta\gamma} \not\subseteq S DRS_\beta^{\beta\gamma} . \text{ Cela peut arriver lorsque } L_\alpha \text{ et } L_\eta \text{ n'ont pas un arrière-plan approprié pour « bien » suivre le dialogue qui a lieu } \textit{via} \text{ la chaîne } v_{\beta\gamma} ;$$

2. seul un des locuteurs concernés est le même dans les deux dialogues :

$$\text{equals}(L_\beta, L_\varphi) \vee \text{equals}(L_\gamma, L_\psi) \vee \text{equals}(L_\beta, L_\psi) \vee \text{equals}(L_\gamma, L_\varphi).$$

On suppose par exemple que $\text{equals}(L_\beta, L_\varphi)$. Ainsi, le cas (II.2) précisé ci-dessus devient :

$S DRS_\alpha^{\beta\gamma} \cap S DRS_\eta^{\beta\psi} \neq \emptyset$, c'est-à-dire, L_α et L_η observent tous les deux que L_β produit des tours de parole destinés en même temps à L_γ et à L_ψ . Cette situation est illustrée dans la figure 5.2.

Un sous-cas intéressant de ce dernier cas présenté ci-dessus a lieu lorsque L_α et L_η se partagent *totalemment* leurs perspectives :

$$\text{equals}(S DRS_\alpha^{\beta\gamma}, S DRS_\eta^{\beta\gamma}) \wedge \text{equals}(S DRS_\alpha^{\beta\psi}, S DRS_\eta^{\beta\psi}).$$

Cette situation est réductible au *premier* sous-cas dans (II.2), afin de satisfaire les deux contraintes précisées ci-dessus ; ensuite, une fois que ces contraintes sont satisfaites, nous avons que :

$$S DRS_\alpha^{\beta\gamma} \cap S DRS_\eta^{\beta\psi} \neq \emptyset \Leftrightarrow S DRS_\alpha^{\beta\gamma} \cap S DRS_\alpha^{\beta\psi} \neq \emptyset.$$

Dans ce cas, du fait qu'il n'y a qu'un locuteur dans deux dialogues différents, il résulte que :

$$S DRS_\alpha^{\beta\gamma} \cap S DRS_\alpha^{\beta\psi} \neq \emptyset \Leftrightarrow \begin{pmatrix} \beta - \gamma \\ \beta - \psi \end{pmatrix},$$

selon la perspective du locuteur L_α .

5.3 Dialogue homme-machine à plusieurs locuteurs

5.3.1 Unicité des points de vue

Dans le paragraphe précédent nous avons considéré plusieurs perspectives sur les dialogues qui ont lieu entre les locuteurs. Dans ce paragraphe, nous n'allons considérer que la perspective d'un seul locuteur, la machine.

Ainsi, en supposant qu'il y a N locuteurs L_1, \dots, L_N et la machine M , la réunion des structures rhétoriques calculées par la machine, pour les dialogues qui ont lieu entre des paires de locuteurs, devient :

$$SDRS_{1NM} = \left(\bigcup_{i=1}^N SDRS_M^{Mi} \right) \cup \left(\bigcup_{i=1}^N \left(\bigcup_{j=i+1}^N SDRS_M^{ij} \right) \right).$$

De plus, les notations utilisées dans le paragraphe précédent prennent la forme suivante :

$$SDRS_{\alpha}^{\alpha\beta} \mapsto SDRS_M^{M\beta}; SDRS_{\alpha}^{\beta\gamma} \mapsto SDRS_M^{\beta\gamma}.$$

Par conséquent, les cas I. et II. spécifiés dans le paragraphe précédent deviennent :

I'. Perspective « impliquée » de la machine : Pour les doubles (M, L_{β}) et (M, L_{φ}) , il y a deux cas :

1. $SDRS_M^{M\beta} \cap SDRS_M^{M\varphi} = \emptyset$. Cela représente une situation de dialogue multi-session, où M maintient des structures rhétoriques séparées pour chaque interlocuteur humain. Ce cas n'est pas vraiment intéressant d'un point de vue rhétorique, mais pose des défis importants au niveau du contrôle de dialogue [123] (notamment la gestion des *conflicts* entre des utilisateurs qui souhaitent l'accès au même moment à une même ressource) ; un exemple dans ce sens, impliquant la machine M et deux locuteurs L_{β} et L_{φ} est montré ci-dessous (les indices numériques spécifient des tours de parole) :

Session 1 :

M_1 : Bonjour, Monsieur, qu'est-ce que je peux faire pour vous ?

L_{β}^1 : Bonjour, je voudrais des DVD sur les éruptions du Kilimanjaro...

M_2 : OK... un instant... Voilà : J'en ai trouvé un, sur l'éruption du Kilimanjaro, qui a eu lieu il y a une centaine de milliers d'années. Je la mets sur votre carte ?

L_{β}^2 : Oui, s'il vous plaît ! Merci beaucoup, au revoir !

Session 2 :

M_1 : Bonjour, Monsieur, qu'est-ce que je peux faire pour vous ?

L_{φ}^1 : Salut ! Eh ben, je voudrais quelque chose sur... le volcan Kilimanjaro... Pourriez-vous me donner un truc avec des photos ?

M_2 : Un instant... Eh ben, j'ai trouvé un dépliant touristique, il y a beaucoup de photos sympa dedans...

L_{φ}^2 : En fait, je voudrais plutôt un film ou quelque chose... Je ne sais pas, est-ce que vous avez un DVD sur le Kilimanjaro ?

M_3 : Eh ben, en fait, on en a un, mais un autre client vient de le prendre. Il sera disponible dans deux semaines...

L_φ^3 : D'accord, pourriez-vous donc me le mettre sur ma liste de réservations, s'il vous plaît ?

M_4 : Bien sûr ! Voilà, c'est fait ! En attendant, est-ce que vous désirez autre chose ? Par exemple, on a un autre DVD, sur les volcans en Afrique, il y a une rubrique sur le Kilimanjaro aussi !

L_φ^4 : Ok, nickel, mettez-le sur ma carte maintenant, s'il vous plaît ! Merci, au revoir !

2. $S DRS_M^{M\beta} \cap S DRS_M^{M\varphi} \neq \emptyset$. Ceci est un dialogue à plusieurs locuteurs, où M est impliquée dans des dialogues simultanés avec les utilisateurs L_β et L_φ . Une généralisation de deux locuteurs (L_β et L_φ), à N locuteurs (L_1 à L_N) est immédiate. Nous présentons ci-dessous un exemple d'un dialogue multi-locuteurs typique pour cette situation, où trois clients humains interagissent avec une machine autour de la tâche de réservations de livres dans une bibliothèque :

M_1 : Bonjour, qu'est-ce que je peux faire pour vous ?

L_φ^1 : Salut, eh ben, je voudrais un livre ou un truc comme ça sur le théâtre français moderne...

M_2 : Pourriez-vous donner plus de précisions, par exemple sur le thème ou sur l'auteur ?

L_φ^2 : Eh ben, un truc du dixseptième siècle... comme Molière par exemple...

L_β^2 : En fait, nous voudrions quelque chose sur les joueurs... la psychologie du joueur...

M_3 : Donc, vous voulez une pièce de théâtre du dixseptième siècle sur les jeux de fortune et la psychologie des joueurs, c'est ça ?

L_β^3 : Oui, c'est ça !

L_φ^3 : En fait, ça ne doit pas forcément être Molière, je veux dire quelque chose de plus léger... ça ira également...

M_4 : Ok, voilà ce que j'ai trouvé pour vous : « Le joueur » de Regnard, écrit en 1696. Est-ce que ça vous ira ?

L_β^4 : Eh ben, de quoi s'agit-il dans cette pièce ?

L_φ^4 : Est-elle une pièce moraliste, avec un joueur qui finit mal, à cause de son vice ?

M_5 : Oui, il s'agit d'un joueur, Valère, qui joue l'amour de sa maîtresse, et finit seul et en faillite. Est-ce que ça vous ira ?

L_φ^5 : Ca a l'air moraliste pour moi...

L_β^5 : Et c'est justement cela que nous cherchons : du théâtre français moraliste du dixseptième siècle, sur des vices tels que les jeux de fortune !

M_6 : Donc, devrais-je la mettre sur vos cartes ? Est-ce qu'une seule copie vous suffira, ou vous désirez plutôt deux copies ?

L_β^6 : Je voudrais un exemplaire pour moi ! Vous pouvez le mettre sur ma carte !

M_7 : Ok, donc je mets une copie séparée pour chacun d'entre vous. Est-ce bien comme ça ?

L_φ^7 : Oui, c'est parfait, merci, au revoir !

L_β^7 : Ouais, merci, à bientôt !

Dans ce deuxième cas (I'. 2.), comme dans le paragraphe précédent, nous pouvons écrire :

$$SDRS_M^{M\beta} = \left(\bigcup_t \text{tour de parole } (\pi(\sigma(M, \beta)) : \text{equals}(t(\pi), t)) \right) \cup \\ \left(\bigcup_{t < t'} \text{tours de parole } (\rho(\pi(\sigma(M, \beta)), \pi'(\sigma(M, \beta))) : \text{equals}(t(\pi), t) \wedge \text{equals}(t(\pi'), t')) \right).$$

Par conséquent :

$$SDRS_M^{M\beta} \cap SDRS_M^{M\varphi} \neq \emptyset \Leftrightarrow \begin{pmatrix} M - \beta \\ M - \varphi \end{pmatrix} \vee \begin{pmatrix} \beta - M \\ \beta - \varphi \end{pmatrix} \vee \begin{pmatrix} \varphi - M \\ \varphi - \beta \end{pmatrix}.$$

Mais :

$$(i) \begin{pmatrix} M - \beta \\ M - \varphi \end{pmatrix} \Rightarrow SDRS_M^{M\beta} \cap SDRS_M^{M\varphi} \neq \emptyset;$$

$$(ii) \begin{pmatrix} \beta - M \\ \beta - \varphi \end{pmatrix} \Rightarrow SDRS_M^{\beta\varphi} \cap SDRS_M^{M\beta} \neq \emptyset;$$

$$(iii) \begin{pmatrix} \varphi - M \\ \varphi - \beta \end{pmatrix} \Rightarrow SDRS_M^{\beta\varphi} \cap SDRS_M^{M\varphi} \neq \emptyset.$$

Les cas (ii) et (iii) sont inclus dans le cas (II'), présenté ci-dessous, donc la seule situation intéressante ici est celle où $\begin{pmatrix} M - \beta \\ M - \varphi \end{pmatrix}$, c'est-à-dire, M produit un énoncé destiné en même temps à L_β et à L_φ .

II'. Perspective « extérieure » de la machine : Pour les triples (M, L_β, L_γ) et (M, L_φ, L_ψ) , où l'ordinateur n'est qu'un observateur des dialogues qui ont lieu entre des locuteurs différents, il y a deux cas :

1. $SDRS_M^{\beta\gamma} \cap SDRS_M^{\varphi\psi} = \emptyset$. Dans ce cas les perspectives de M sur les dialogues qui ont lieu via les chaînes $\nu_{\beta\gamma}$ et $\nu_{\varphi\psi}$ n'ont rien à voir une avec l'autre. En fait, la machine observe deux dialogues indépendants qui ont lieu en parallèle ; ce cas peut être utile pour régler les énoncés de M , destinés à un locuteur $L_\lambda \in \{L_\beta, L_\gamma, L_\varphi, L_\psi\}$ (voir le paragraphe suivant pour plus de détails concernant cette situation) ;
2. $SDRS_M^{\beta\gamma} \cap SDRS_M^{\varphi\psi} \neq \emptyset$. Dans ce cas M observe deux dialogues qui ont quelques tours de parole en commun³ Ici, il y a deux sous-cas « raisonnables », pas en contradiction avec le sens commun :

(a) $\text{equals}((L_\beta, L_\gamma), (L_\varphi, L_\psi))$. Il s'agit d'un sous-cas trivial, puisqu'il implique que :

$$SDRS_M^{\beta\gamma} \cap SDRS_M^{\beta\varphi} \neq \emptyset;$$

(b) $\text{equals}(L_\beta, L_\varphi) \vee \text{equals}(L_\gamma, L_\psi) \vee \text{equals}(L_\beta, L_\psi) \vee \text{equals}(L_\gamma, L_\varphi)$. Pour simplifier la présentation, nous supposons que $\text{equals}(L_\beta, L_\varphi)$, ce qui revient à :

$$SDRS_M^{\beta\gamma} \cap SDRS_M^{\beta\varphi} \neq \emptyset \Leftrightarrow \begin{pmatrix} \beta - \gamma \\ \beta - \varphi \end{pmatrix}, \text{ selon la perspective de la machine.}$$

³Le lecteur peut se reporter au paragraphe 5.2 pour voir à quoi revient le fait que deux structures de discours ont quelque chose en commun.

Etant donné que ces dernières situations concernent des dialogues entre des locuteurs humains et n'impliquent pas directement la machine, nous ne donnerons pas d'exemples ici ; il est facile de s'imaginer des dialogues entre des locuteurs humains, où la machine ne fait qu'écouter.

5.3.2 Situations retenues en dialogue homme-machine à plusieurs locuteurs

Toutes les situations de dialogue multi-locuteurs que nous avons présentées dans ce chapitre doivent être détaillées davantage, jusqu'à un niveau algorithmique, qui ouvre la voie vers des implémentations dans des systèmes de dialogue réels. Ainsi, nous allons spécifier tout d'abord le mécanisme de mise à jour des structures rhétoriques, pour la situation où la machine est en train de *générer* un tour de parole. Nous nous limitons à cette situation pour rester fidèles aux buts de cette thèse, qui sont de proposer une modélisation des contraintes pragmatiques en génération ; nous croyons cependant que ce cadre est applicable aussi à l'interprétation pragmatique, pour étendre par exemple les travaux de Caelen et Xuereb [35]. Plus précisément, nous allons spécifier le mécanisme de mise à jour pour des structures rhétoriques du type $(SDRS_M^{Mi})_{i=1,\dots,N}$, en prenant en compte les structures du type $(SDRS_M^{ij})_{i \neq j; i, j=1,\dots,N}$.

Ainsi, nous considérons deux situations de dialogue orienté service, où la machine et plusieurs (N) locuteurs (utilisateurs) participent :

A. Dialogue orienté service où la machine (par exemple, un bibliothécaire ou un vendeur de billets de train) parle simultanément à plusieurs clients qui ne parlent pas entre eux ; ce type de dialogue est aussi approprié dans un contexte d'enseignement, par exemple. Dans ce dernier type d'interaction, un locuteur est l'enseignant, tandis que les autres locuteurs ne parlent à personne, sauf à l'enseignant, qui, à son tour, prend en compte les connaissances d'un, plusieurs, ou de tous ses interlocuteurs dans ses interventions.

Formellement, la perspective rhétorique de M (c'est-à-dire, les SDRS que celle-ci calcule) sur le contexte discursif est :

$$SDRS_{1NM} = \bigcup_{i=1}^N SDRS_M^{Mi}.$$

Ce contexte discursif est mis à jour lorsque la machine produit un tour de parole. Comme nous l'avons vu dans les chapitres précédents, ce tour est d'abord calculé sous forme logique par le contrôleur de dialogue, pour qu'ensuite le générateur, en prenant en compte les contraintes pragmatiques abordées dans cette thèse, produise une forme linguistique. La forme logique prototypique d'un tel tour de parole est :

$$\tilde{K}(\pi(M, I)) = \text{Act}(\bigwedge_{i \in I} \text{dest}(L_i) \wedge K(\pi(M, I))).$$

Dans cette expression, **Act** représente une fonction qui attribue un type d'acte de langage (selon la taxinomie introduite dans le chapitre 3) à un contenu propositionnel (cf. aussi chapitre 7, où cette fonction est notée par F) ; l'énoncé ainsi spécifié sous forme logique est destiné aux locuteurs L_i pour $i \in I$, où $I \subseteq \{1, \dots, N\}$. Le prédicat **dest/1** est vrai si et seulement si l'énoncé libellé π est destiné au locuteur dont l'identifiant est l'argument de ce prédicat. Ainsi, dans notre modèle le fait qu'un énoncé est destiné à un ensemble de locuteur fait partie de sa sémantique⁴ ; cela n'implique cependant pas que le fait que l'énoncé soit *perçu* ou entendu par un locuteur

⁴Evidemment, il s'agit de considérer ici la notion de « sémantique » dans un sens élargi par rapport aux conventions adoptées dans les chapitres antérieurs de cette thèse, ou à l'acception usuelle de ce terme.

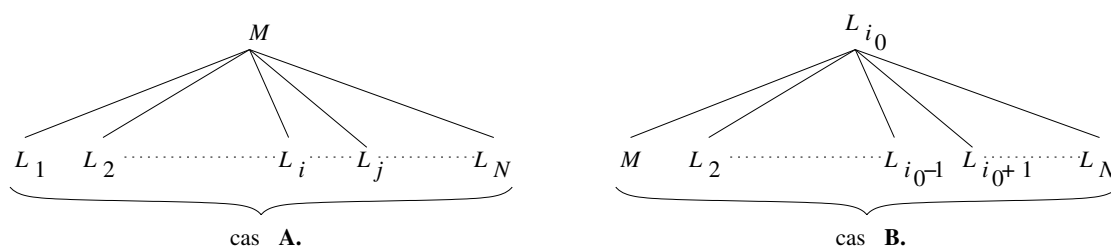


FIG. 5.3 – Situations pertinentes en dialogue homme-machine à plusieurs locuteurs

fasse partie de cette sémantique. La sémantique de l'énoncé spécifie seulement l'ensemble des *destinataires* ; le fait que ceux-ci reçoivent vraiment l'énoncé est un autre problème, dont les effets ne peuvent apparaître qu'*a posteriori* par rapport à la production de l'énoncé.

B. Dialogue orienté service (comme dans le cas **A.** ci-dessus), où la machine est un client (ou un étudiant dans les dialogues d'enseignement). L'essentiel est que dans ce cas la machine ne parle qu'à un locuteur unique (par exemple, le bibliothécaire ou l'enseignant), tout en prenant cependant en compte les conversations entre ce locuteur (auquel la machine parle) et les autres locuteurs impliqués dans la conversation à plusieurs.

Formellement, M se construit une représentation discursive du type suivant, sur ce type de dialogue :

$$SDRS_{1NM} = SDRS_M^{Mi_0} \cup \left(\bigcup_{j=1; j \neq i_0}^N SDRS_M^{i_0j} \right).$$

Dans cette expression, L_{i_0} est le locuteur « central » auquel la machine (et tout le monde) parle ; ce locuteur offre en fait le service qui circonscrit le dialogue (il est donc par exemple le bibliothécaire virtuel).

Le contexte discursif dans ce type de dialogues est mis à jour en ajoutant à la structure rhétorique des formules logiques (qui correspondent aux intentions communicationnelles de la machine) du type suivant :

$$\tilde{K}(\pi(M, i_0)) = \text{Act}(\text{dest}(L_{i_0}) \wedge K(\pi(M, i_0))).$$

Ici, l'ensemble I des destinataires de l'énoncé π se réduit à un seul élément, i_0 , pour le locuteur « central » L_{i_0} .

La formule logique $\tilde{K}(\pi(M, i_0))$ doit être ajoutée à la structure rhétorique $SDRS_M^{Mi_0}$, en prenant en compte les structures rhétoriques $SDRS_M^{i_0j}$, $j \in \{1, N\} \setminus \{i_0\}$. C'est justement à cette prise en compte du contexte que nous allons dédier la plus grande partie du paragraphe suivant.

Ces deux situations de dialogue particulières sont illustrées dans la figure 5.3. Il est utile à établir une correspondance entre ces situations particulières et les cas plus généraux présentés dans le paragraphe antérieur : dans le cas **A.**, la machine est toujours un locuteur « impliqué », donc en situation (I'), tandis que dans le cas **B.**, la machine est soit un locuteur impliqué, soit un locuteur en train de faire la transition du statut d'« observateur » vers celui de locuteur « impliqué » – lorsque le locuteur « central » vient d'être en train de parler à un autre locuteur. Par conséquent, dans ce cas la machine est soit également en situation (I'), soit en train de faire une transition de la situation (II') vers la situation (I').

5.3.3 Mise à jour de la structure de discours dialogique

Afin de spécifier de manière algorithmique le processus de mise à jour de la structure rhétorique en dialogue multi-locuteurs, nous nous concentrons seulement sur le cas où la machine est en train de produire un tour de parole, qui est censé être ajouté aux structures rhétoriques concernées. Nous limitons ainsi notre description en essayant de rester fidèles aux buts de cette thèse, notamment d'établir et formaliser les contraintes pragmatiques sur la *verbalisation* d'une intention communicationnelle. De plus, nous nous limitons pour l'instant aux cas **A.** et **B.**, présentés dans le paragraphe précédent. Comme nous le verrons plus loin, cette dernière limitation n'est qu'apparente, parce que, nous allons le montrer, chaque dialogue orienté service peut être décomposé dans une suite alternée de dialogues du type **A.** et **B.**. Ainsi :

Cas A. Les structures rhétoriques à mettre à jour sont :

$$SDRS_{1NM} = \bigcup_{i=1}^N SDRS_M^{Mi}.$$

La formule logique (fournie par le contrôleur de dialogue) censée être ajoutée à ces structures de discours est notée par $\tilde{K}(\pi(M, I))$ et a l'expression précisée dans le paragraphe antérieur.

Etant donnés ces éléments, nous proposons un algorithme pour la mise à jour des structures rhétoriques pour les dialogues du type **A.** :

pour une intention communicationnelle $\tilde{K}(\pi(M, I))$:

1. trouver l'ensemble I des destinataires de cette intention :

$$I = \bigcup_{\text{equals}(\text{dest}(L_i), 1)} \text{arg}(\text{dest}(L_i));$$

informellement, cette expression dit que l'ensemble I est constitué de l'union de indices des identités des destinataires ; ces identités des destinataires sont récupérées à leur tour comme les arguments des instanciations des prédicats $\text{dest}/1$ dans l'intention communicationnelle ;

2. choisir les structures rhétoriques à mettre à jour :

$$\bigcup_{i \in I} SDRS_M^{Mi};$$

informellement, cette expression dit que les structures à mettre à jour sont données par les éléments de l'ensemble I des destinataires de l'intention communicationnelle : il ne faut mettre à jour que les SDRS que la machine se construit sur les dialogues qu'elle agence avec les locuteurs dont les identités sont précisées par les éléments de l'ensemble I ;

3. extraire le contenu sémantique de l'intention communicationnelle :

$$(a) \text{Act}(\bigwedge_{i \in I} \text{dest}(L_i) \wedge K(\pi(M, I))) \mapsto \text{Act}(\bigwedge_{i \in I} \text{dest}(L_i)) \wedge \text{Act}(K(\pi(M, I))) \mapsto \bigwedge_{i \in I} \text{Act}(\text{dest}(L_i)) \wedge \text{Act}(K(\pi(M, I)));$$

informellement, cette expression dit que nous pouvons distribuer les fonctions du type $\text{Act}/1$, parce qu'il s'agit du même acte (« faire-faire », « faire-savoir », « faire-faire-savoir », « faire-pouvoir », ou « faire-devoir » – cf. chapitre 3) qui est réalisé par *toute* l'intention communicationnelle : par exemple, pour informer deux interlocuteurs L_i et L_j de la vérité d'une proposition α , le contrôleur de dialogue produit une intention communicationnelle du type $F^S(\text{dest}(L_i) \wedge \text{dest}(L_j) \wedge K(\alpha))$, ce qui est équivalent à informer les locuteurs L_i et L_j que nous leur destinons un énoncé, α , et que nous voulons les *informer* du contenu de cet énoncé, $K(\alpha)$, ce qui revient à

réaliser trois actes : $F^S(\text{dest}(L_i)) \wedge F^S(\text{dest}(L_j)) \wedge F^S(K(\alpha))$. C'est comme si, au lieu de destiner de manière implicite l'énoncé α à L_i et L_j on produisait trois énoncés, du type : ' L_i , je vais vous communiquer quelque chose', ' L_j , je vais vous communiquer la même chose qu'à L_i ', et 'cette chose est que α '.

- (b) $\bigwedge_{i \in I} \text{Act}(\text{dest}(L_i)) \wedge \text{Act}(K(\pi(M, I))) \vee \neg \bigwedge_{i \in I} \text{Act}(\text{dest}(L_i)) \mapsto$
 $\bigwedge_{i \in I} (\text{Act}(\text{dest}(L_i)) \vee \neg \text{Act}(\text{dest}(L_i))) \wedge \text{Act}(K(\pi(M, I))) \mapsto \text{Act}(K(\pi(M, I)))$;

informellement, cette expression dit que, une fois isolés les prédicats qui spécifient les destinataires de l'intention communicationnelle, nous pouvons les éliminer, en composant cette intention avec les négation de ces prédicats, pour ainsi ne rester qu'avec le contenu proprement dit sémantique de l'énoncé ;

- (c) $\text{Act}(K(\pi(M, I))) \mapsto \text{Act}(K(\pi))$;

informellement, cette expression dit que nous supprimons même les informations sur l'émetteur de l'énoncé π , puisque cette information est implicitement contenue dans le fait que l'algorithme, qui ne se déclenche que lorsqu'un énoncé est en train d'être produit par la machine, tourne ; nous préparons ainsi l'entrée dans la procédure de mise à jour des structures rhétoriques en dialogue classique à deux locuteurs ;

4. **pour** $i \in I$: `simple_update` ($\text{Act}(K(\pi))$, $S DRS_M^{Mi}$).

Dans cet algorithme, la fonction `simple_update/2` met à jour une structure rhétorique (précisée comme son deuxième argument), avec une formule logique (précisée comme son premier argument), dans les cas du dialogue classique entre deux locuteurs (dont un est la machine). Cette fonction opère donc une mise à jour des structures rhétoriques selon les principes présentés dans les chapitres 3 et 4 de cette thèse.

Cas B. La structure rhétorique que la machine doit mettre à jour dans ce cas est :

$$\overline{SDRS}_{1NM} = S DRS_M^{Mi_0}.$$

En fait, $S DRS_M^{Mi_0}$ est mise à jour avec le contenu sémantique $\text{Act}(K(\pi))$, extrait de l'intention communicationnelle $\tilde{K}(\pi(M, i_0))$. Cette mise à jour se fait en prenant en compte les structures rhétoriques $(S DRS_M^{i_0j})_{j=1, \dots, N; j \neq i_0}$. Autrement dit, l'acte de langage censé être généré couramment est relié aux énoncés dans la SDRS censée être mise à jour, tout en prenant en compte les autres énoncés produits en dialogue, énoncés qui ont un rapport soit avec l'énoncé ajouté à la SDRS, soit avec les énoncés auxquels cet énoncé peut être connecté, selon les sémantiques des relations rhétoriques et les principes décrits dans les chapitres 3 et 4. Ces autres énoncés sont, à leur tour, produits dans des dialogues entre les autres locuteurs (sauf la machine) et le locuteur « central ». Ainsi, si un énoncé dans la SDRS concernée ($S DRS_M^{Mi_0}$) est relié *via* une relation rhétorique de **contradiction** (cf. chapitres 6 et / ou 7 pour l'ensemble des relations rhétoriques de contradiction : *P-Corr*, *Background_q*, *NEI*, *Alternation* et *Contrast*), à un énoncé *ultérieur* dans une autre structure rhétorique (du type type $S DRS_M^{i_0j}$), alors l'acte censé être généré peut être connecté au premier énoncé (celui dans $S DRS_M^{Mi_0}$) d'une des manières suivantes :

- via une relation rhétorique de contradiction, par rapport à l'énoncé dans $S DRS_M^{i_0j}$, suivie par une relation rhétorique de **confirmation** (cf. également chapitres 6 ou 7 pour l'ensemble des relations rhétoriques de confirmation : *Q-Elab*, *P-Elab*, *Elab_q*, *Narration_q*,

QAP, ACK, IQAP, Background, Consequence, Elaboration, Narration et Parallel), par rapport à l'énoncé dans $SDRS_M^{Mi_0}$, en insérant dans la structure $SDRS_M^{Mi_0}$, le fragment de la structure $SDRS_M^{i_0j}$ qui a réalisé la contradiction avec le premier énoncé, pourvu que cet énoncé n'ait pas été assumé par son émetteur (voir chapitre 7 pour une définition précise des engagements – le fait qu'un énoncé soit assumé par un locuteur est équivalent au fait que ce locuteur s'engage à cet énoncé); essentiellement, un énoncé est assumé par un locuteur soit si ce locuteur l'affirme, soit si un autre locuteur l'affirme, mais le locuteur concerné exprime (de manière explicite ou implicite – c'est-à-dire, par exemple, en élaborant sur cet énoncé) son accord à l'égard de cet énoncé;

- via une relation rhétorique d'élaboration ($P - Elab$, $Elab_q$ ou $Q - Elab$ en dialogue, ou *Elaboration* en monologue), par rapport à une éventuelle suite d'énoncés dans les structures du type $SDRS_M^{i_0j}$, énoncés qui élaborent à leur tour sur l'énoncé concerné, situé dans $SDRS_M^{Mi_0}$;
- par défaut, s'il n'y a aucune contrainte déterminée par les structures rhétoriques du type $SDRS_M^{i_0j}$, sur l'énoncé concerné, dans $SDRS_M^{Mi_0}$.

Etant données ces consignes, nous proposons ci-dessous un algorithme qui les prend en compte, pour mettre à jour la structure rhétorique pour les dialogues du type **B.** :

pour une intention communicationnelle $\tilde{K}(\pi(M, i_0))$:

1. identifier la structure rhétorique à mettre à jour : $SDRS_M^{Mi_0}$;
2. trouver les structures rhétoriques à prendre en compte dans le processus de mise à jour : $(SDRS_M^{i_0j})_{j \notin \{i_0, M\}}$;
3. **pour** chaque énoncé π_k qui se trouve déjà dans $SDRS_M^{Mi_0}$:
 - (a) **pour** tout $j \in \{1, \dots, N\} \setminus \{i_0\}$:
 - i. vérifier s'il existe un énoncé π'_k qui n'a pas été produit par $emitter(\pi_k)$, mais qui se trouve dans $SDRS_M^{i_0j}$, tel qu'il existe une relation rhétorique $\rho(\pi_k, \pi'_k)$ aussi dans $SDRS_M^{i_0j}$:
 - A. **si** oui (ρ existe), **alors** :
 - A1) **si** ρ est une relation de contradiction, **alors** :
 - I) vérifier s'il existe un énoncé π''_k qui n'a pas été produit par $emitter(\pi'_k)$, mais qui se trouve dans $SDRS_M^{i_0j}$, tel qu'il existe une relation rhétorique $\rho'(\pi'_k, \pi''_k)$ dans $SDRS_M^{i_0j}$:
 - I.1) **si** oui, **alors** :
 - I.1.1) **si** ρ' est une relation rhétorique de confirmation, **alors** marquer π_k comme un énoncé *non-candidat* pour π (ce qui implique le fait qu'aucune relation rhétorique ne sera calculée entre π et π_k) ;
 - I.1.2) **sinon, alors** garder π_k dans la liste des énoncés candidats⁵ pour π ;

⁵Un énoncé π_c est un « énoncé candidat » pour un autre énoncé π si et seulement si l'algorithme essayera de calculer une relation rhétorique entre π_c et π . Il faut aussi noter que π_c est temporellement antérieur à π .

I.2) **sinon** continuer avec l'étape I., en itérant sur les indices $k'' = \arg(\pi_k'' : \pi_k'' \in SDRS_M^{i_0j} \wedge \neg \text{equals}(\text{emitter}(\pi_k''), \text{emitter}(\pi_k')))$;

II) mettre à jour la liste des énoncés non-candidats pour $\pi(M, i_0)$, dans $SDRS_M^{Mi_0}$:

$$NC(\pi(M, i_0))^{SDRS_M^{Mi_0}} \leftarrow NC(\pi(M, i_0))^{SDRS_M^{Mi_0}} \cup \{K(\pi_k)\} ;$$

A2) **sinon, si** ρ est une relation d'élaboration (cf. l'ensemble des relations d'élaboration précisées ci-dessus), **alors** :

I) marquer $K(\pi_k) \wedge K(\pi_k')$ comme un énoncé candidat pour $\tilde{K}(\pi(M, i_0))$;

II) mettre à jour la liste des candidats *composés*⁶ :

$$CC(\pi(M, i_0))^{SDRS_M^{Mi_0} \cup SDRS_M^{i_0j}} \leftarrow$$

$$CC(\pi(M, i_0))^{SDRS_M^{Mi_0} \cup SDRS_M^{i_0j}} \cup \{\pi_k, K(\pi_k'), \wedge_{\rho} \rho(\sigma(\pi_k, \pi_k'))\} ;$$

dans cette structure nous gardons l'étiquette de l'énoncé π_k , accompagné par la *sémantique* de l'énoncé π_k' et par les *étiquettes* des relations rhétoriques qui relient ces deux énoncés entre eux ; cela est nécessaire pour pouvoir reconstituer une sorte de « trace » sur les énoncés, en ce que nous retenons la *sémantique* de l'énoncé le plus récent, aussi que l'étiquette de l'énoncé plus ancien (qui agit comme un point d'accroche au contexte de dialogue), et les relations rhétoriques qui les relient (les *sémantiques* de ces relations rhétoriques sont aisément récupérées à partir de leurs types, parce que chaque type de relation rhétorique est défini par une *sémantique* unique – cf. chapitre 3) ;

A3) **sinon, alors** garder π_k comme un énoncé candidat (dans le sens précisé ci-dessus) pour π ;

B. **sinon** (ρ n'existe pas), **alors** itérer sur $k' = \arg(\pi_k' : \pi_k' \in SDRS_M^{i_0j})$;

cela revient à chercher d'autres énoncés π_k' dans la SDRS que la machine calcule pour le dialogue qui a lieu entre L_{i_0} (le locuteur « central ») et un locuteur L_j

ii. retourner les listes :

$$NC(\pi(M, i_0); j)^{SDRS_M^{Mi_0}} \leftarrow NC(\pi(M, i_0))^{SDRS_M^{Mi_0}} ;$$

$$CC(\pi(M, i_0); j; k)^{SDRS_M^{Mi_0}} \leftarrow CC(\pi(M, i_0))^{SDRS_M^{Mi_0} \cup SDRS_M^{i_0j}} ;$$

(b) calculer les listes globales des énoncés non-candidats et des candidats composés pour $\pi(M, i_0)$:

$$NC(\pi(M, i_0)) = \bigcup_j NC(\pi(M, i_0); j)^{SDRS_M^{Mi_0}} ;$$

$$CC(\pi(M, i_0); k) = \bigcup_j CC(\pi(M, i_0); j; k)^{SDRS_M^{Mi_0}} ;$$

(c) mettre à jour la structure rhétorique à laquelle l'énoncé courant $\pi(M, i_0)$ est censé être ajouté ; pour cela, éliminer les énoncés non-candidats pour $\pi(M, i_0)$ dans la SDRS initiale :

⁶Par « candidat composé » nous entendons un énoncé candidat qui est composé *via* une conjonction entre deux énoncés précédents, accompagné par les étiquettes des relations rhétoriques qui relient ces deux énoncés précédents « constitutifs », par l'étiquette de l'énoncé le plus ancien, et par la *sémantique* de l'énoncé le plus récent dans cette conjonction.

$$SDRS_M^{Mi_0} \leftarrow SDRS_M^{Mi_0} \setminus NC(\pi(M, i_0));$$

(d) calculer la SDRS *composée* :

$SDRS_M^{Mi_0}(k) \leftarrow SDRS_M^{Mi_0} \cup CC(\pi(M, i_0); k)$; pour chaque k en tant que deuxième argument de $CC(\pi(M, i_0); k)$ il existe un énoncé (ou constituant discursif) π_k dans $SDRS_M^{Mi_0}$, puisque si un énoncé avait été supprimé (comme un non-candidat pour $\pi(M, i_0)$), alors il n'aurait pas été un membre dans la composition avec une autre SDRS ;

essentiellement, à cette étape nous ajoutons à la SDRS censée être mise à jour, la liste des candidats composés d'un énoncé π_k qui se trouve déjà dans cette SDRS (cf. l'étape 3 dans cet algorithme) ;

4. calculer la structure rhétorique censée être mise à jour avec l'énoncé $\pi(M, i_0)$ que la machine est en train de générer :

$$SDRS_M^{Mi_0} \leftarrow \bigcup_{k:\pi_k \in SDRS_M^{Mi_0}} SDRS_M^{Mi_0}(k);$$

essentiellement, dans cette étape nous réunissons toutes les SDRS composées, calculées à l'étape 3.(d), selon les énoncés π_k qui se trouvaient déjà dans la SDRS d'origine, celle que nous devons mettre à jour *via* la procédure présentée ici ;

5. extraire le contenu propositionnel (c'est-à-dire, la sémantique proprement dite) de l'intention communicationnelle censée être générée par la machine (cf. l'étape 3. dans l'algorithme pour la mise à jour des structures rhétoriques pour les dialogues du type **A.** – ici nous avons simplifié la présentation par rapport à l'algorithme antérieur) :

$$\tilde{K}(\pi(M, i_0)) \mapsto \text{Act}(K(\pi));$$

6. appeler la procédure (qui fonctionne selon les principes décrits dans les chapitres 3 et 4) pour la mise à jour des structures rhétoriques en dialogue classique, bi-locuteur :

$$\text{simple_update}(\text{Act}(K(\pi)), SDRS_M^{Mi_0}).$$

Dans les deux situations de conversation **A.** et **B.**, le problème essentiel relevant du dialogue multi-locuteurs revient à choisir la (les) structure(s) rhétorique(s) à mettre à jour : dans la situation **A.**, le choix est piloté par l'ensemble des destinataires de l'intention communicationnelle censée être générée par la machine, tandis que dans la situation **B.**, le choix est piloté par la représentation que la machine se construit sur les dialogues que le destinataire de l'énoncé a avec les autres locuteurs. En conséquence, tandis que dans la situation **A.** le choix des structures rhétoriques est plutôt évident, et fixé par le contrôleur de dialogue (via les prédicats *dest/1* dans la formule logique correspondante à l'énoncé censé être généré), dans la situation **B.** la structure rhétorique à mettre à jour est *construite* de manière itérative, selon la progression du dialogue et en prenant en compte les dialogues que les autres locuteurs ont avec le locuteur auquel la machine s'adresse. Ainsi, dans la situation **B.**, les structures de discours que la machine calcule pour les dialogues établis entre son interlocuteur et les autres locuteurs agissent comme des contraintes supplémentaires sur les points d'attachement possibles pour l'énoncé généré, à la structure rhétorique du dialogue entre la machine et son interlocuteur – le locuteur « central ». Ces contraintes supplémentaires proviennent du fait que la structure rhétorique « globale » que la machine se construit en prenant en compte les dialogues des autres locuteurs contient *a priori* plus d'énoncés et de relations rhétoriques entre ceux-ci, ce qui revient à la nécessité de résoudre éventuellement plus des référents dans la structure rhétorique.

5.4 Calcul de la structure rhétorique : exemple étendu en dialogue à plusieurs

L'algorithme pour la mise à jour des structures rhétoriques, que nous avons présenté *in extenso* dans le paragraphe 5.3, sera mis à l'épreuve sur un dialogue homme-machine à quatre locuteurs, dont trois humains. Le dialogue place les quatre locuteurs (trois humains, notés par L_1 , L_2 et L_3 , et la machine, M) dans les deux situations **A.** et **B.** ; en fait, le long de ce dialogue, qui est, à notre avis, pertinent pour les services du type « bibliothécaire virtuel », les situations **A.** et **B.** se succèdent en alternance.

Pour chaque situation il y a plusieurs cas particuliers, selon les destinataires de chaque tour de parole généré par la machine :

– Cas **A.** :

1. $\tilde{K}(\pi(M, I)) \wedge I = \{1, 2, 3\}$;
2. $\tilde{K}(\pi(M, I)) \wedge I = \{2, 3\}$ (ou n'importe quel double) ;
3. $\tilde{K}(\pi(M, I)) \wedge I = \{3\}$ (ou n'importe quel indice seul) ;

– Cas **B.** :

1. $\tilde{K}(\pi(M, 1)) \wedge \{SDRS_M^{12}, SDRS_M^{13} \text{ prises en compte}\}$;
2. $\tilde{K}(\pi(M, 1)) \wedge \{SDRS_M^{13} \text{ prise en compte}\}$;
3. $\tilde{K}(\pi(M, 1)) \wedge \{\text{aucune autre structure de discours n'est prise en compte}\}$.

Les situations **A.1**, **A.2** et **A.3** sont pilotées par le contrôleur de dialogue (via les prédicats $\text{dest}/1$ dans $\tilde{K}(\pi(M, I))$), tandis que les situations **B.1**, **B.2** et **B.3** sont pilotées par le contexte du dialogue multi-locuteurs, en fait par les conversations qui ont L_1 , L_2 et L_3 comme des participants. Ainsi, nous allons analyser de manière détaillée un dialogue typique entre la machine et trois locuteurs humains. Ici, nous allons aussi montrer comment M peut commuter automatiquement entre les situations **A.** et **B.**, le long du même dialogue.

Le dialogue concerne la recherche et la réservation de livres dans une bibliothèque, où l'ordinateur est le bibliothécaire et L_1 , L_2 et L_3 sont les clients. C'est M qui ouvre le dialogue, en se présentant. Le dialogue est annoté en termes de locuteurs, tours de parole et énoncés, sous la forme suivante :

$$M_j : \langle \text{énoncé} \rangle^{\pi_{j_0}} \dots \langle \text{énoncé} \rangle^{\pi_{j_1}} ;$$

$$L_k^j : \langle \text{énoncé} \rangle^{\pi_{j(i+1)}} \dots \langle \text{énoncé} \rangle^{\pi_{j_2}} .$$

Ici, j est l'indice du tour de parole, k est l'indice de l'identité du locuteur humain, et π_{j_s} désigne le s ème énoncé dans le j ème tour de parole.

Ainsi, le dialogue est listé, sous forme annotée, ci-dessous ; les énoncés de la machine sont générées au fur et à mesure, après chaque mise à jour de la structure rhétorique, mais ce processus n'est pas détaillé ici :

M_1 : Bonjour, je suis Groplan ^{π_{11}} , je peux vous aider à chercher un document dans notre bibliothèque ^{π_{12}} . Vous pouvez à tout moment demander des renseignements concernant la disponibilité d'un certain livre, CD ou DVD ^{π_{13}} . Qu'est-ce que je peux faire pour vous ? ^{π_{14}}

L_1^1 : Bonjour^{π15}, eh ben, nous voudrions une pièce de théâtre, sur un inceste^{π16}.

L_2^1 : Une tragédie ancienne ou un truc comme ça^{π17}.

M_2 : Un instant s'il vous plaît... voilà :^{π21} j'ai trouvé huit tragédies anciennes où un inceste apparaît^{π22}. Pourriez-vous donner plus de précisions, par exemple sur l'auteur ?^{π23}

L_1^2 : Aristophane ?^{π24}

L_3^2 : Ou sinon, Virgile ?^{π25}

M_3 : Il n'y a pas de tragédie ancienne écrite par Aristophane ou Virgile et où un inceste apparaît^{π31}. Néanmoins, j'ai trouvé trois tragédies par Sophocle, une tragédie par Eschyle et une tragédie par Euripide, où un inceste apparaît^{π32}. Voudriez-vous choisir un de ces auteurs, vous renseigner sur d'autres auteurs pertinents, ou effectuer une nouvelle recherche ?^{π33}

L_2^3 : Qu'est-ce que vous avez de Sophocle ?^{π34}

M_4 : Par exemple, « Oedipe roi »^{π41}.

L_1^4 : De quoi s'agit-il dans cette pièce ?^{π42}

L_3^4 : Est-ce qu'il y a un inceste entre frère et sœur ?^{π43}

M_5 : Il n'y a pas d'inceste entre frère et sœur dans « Oedipe roi »^{π51}, mais il s'agit, essentiellement, d'un parricide, suivi d'un inceste entre mère et fils^{π52}.

L_2^5 : Ok, cela a l'air bon !^{π53}

L_1^5 : Donc, c'est comme dans le film « Ma mère » avec Isabelle Huppert^{π54}. J'ai adoré ce film !^{π55}

M_6 : Alors, cette tragédie vous convient ?^{π61}

L_3^6 : Est-ce que vous avez aussi des interprétations sur DVD, pour cette pièce ?^{π62}

L_1^6 : Ouais, une opéra, je crois que cela nous ira à merveille !^{π63}

M_7 : Oui, effectivement, je peux vous donner un opéra avec Luciano Pavarotti dans le rôle de Tirésias, avec Riccardo Muti comme chef d'orchestre, jouée en mars 1974 au théâtre Scala de Milan^{π71}. Est-ce que cela vous ira ?^{π72}

L_1^7 : Oui, Ok pour moi !^{π73}

L_3^7 : Pour moi aussi !^{π74}

L_2^7 : Oui, pour moi c'est bon !^{π75}

M_8 : Donc, je vais vous donner « Oedipe roi » aux éditions Hachette de 1972, et un DVD avec un opéra avec le même titre, jouée au théâtre Scala de Milan, en 1974^{π81}. Est-ce qu'un seul exemplaire de chaque document vous suffira^{π82}, ou peut-être vous désirez une copie séparée pour chacun d'entre vous ?^{π83}

L_3^8 : Je voudrais une copie séparée du livre pour moi^{π84}. Par contre, pour le DVD, nous pourrions le regarder ensemble^{π85}.

L_1^8 : Moi pareil pour le DVD^{π86}, par contre pour le livre, je pourrai le lire avec ma femme (il s'agit de L_2 – n.a.)^{π87}, du coup on pourra avoir tous les deux la même copie du livre^{π88}.

L_2^8 : Oui, je suis d'accord !^{π89}

M_9 : Ok, donc je vous prête une copie pour le DVD et deux copies pour le livre ^{π_{91}} .
Est-ce bien ce que vous désirez ? ^{π_{92}}

L_1^9 : Oui, c'est ça ! ^{π_{93}}

L_3^9 : En effet ! ^{π_{94}}

L_2^9 : Ouais, ouais... Ok ! ^{π_{95}}

M_{10} : Ok ^{π_{101}} , alors, dites-moi, s'il vous plaît, à quel nom je fais les réservations ? ^{π_{102}}

L_1^{10} : Eh ben, je pense qu'on peut mettre une copie du livre sur ma carte, ainsi que le DVD ^{π_{103}} .

L_3^{10} : Et l'autre copie du livre, vous pourriez la mettre sur ma carte ^{π_{104}} .

M_{11} : Ok ^{π_{111}} . Voilà ^{π_{112}} . Je vous remercie d'avoir utilisé notre service, à bientôt, au revoir ! ^{π_{113}} Bonne journée ! ^{π_{114}}

L_2^{11} : Bonne journée à vous aussi ! ^{π_{115}}

L_3^{11} : Au revoir ! ^{π_{116}}

L_1^{11} : Au revoir, à bientôt ! ^{π_{117}}

Ce dialogue combine les situations **A.** et **B.**, parce que M , bien qu'il reste le « serveur » (c'est-à-dire, le bibliothécaire), enregistre les tours de parole *via* lesquels les locuteurs humains se complètent réciproquement afin de raffiner leurs requêtes. Par conséquent, le dialogue reste pour la plupart du temps en situation **B.**, avec de courts intervalles de temps où la machine commute en situation **A.**

Nous allons préciser tout d'abord comment la machine commute entre les deux situations **A.** et **B.** L'information essentielle que la machine utilise pour commuter entre les deux situations réside dans l'existence ou la non-existence des relations MP dans les structures rhétoriques. Cela est spécifié formellement ci-dessous :

pour chaque tour de parole i :

si ($\exists \rho \in SDRS^{(\neg M)}(i) : equals(\rho, MP)$), **alors** :
situation_de_dialogue \leftarrow **B.** ;

sinon, alors :
situation_de_dialogue \leftarrow **A.**

De manière informelle, la situation de dialogue où la machine se place par défaut est **A.** et, si dans le tour de parole courant il y a une relation MP dans une des SDRS qui représente un des dialogues entre les usagers humains, alors la machine commute en situation **B.** Cette décision est prise pour chaque tour de parole. Nous remarquons ainsi l'utilité de cette pseudo-relation rhétorique MP ; nous n'avons même pas besoin de définir une taxinomie plus fine sur ce type de relations, puisque MP ne fait que déclencher la commutation de la machine en situation **B.**

Une fois ces détails réglés, nous pouvons maintenant tracer la mise à jour des structures rhétoriques dans le dialogue montré ci-dessus ; cette mise à jour se fait, bien évidemment, au fur et à mesure que le dialogue progresse :

1. La procédure décrite dans ce chapitre se place par défaut dans la situation de dialogue **A.** et construit une SDRS composée des énoncés π_{11} , π_{12} , π_{13} et π_{14} , et des relations rhétoriques $Elaboration(\pi_{11}, \pi_{12})$, $Consequence(\pi_{12}, \pi_{13})$ et $Background(\pi_{13}, \pi_{14})$;

2. L'interpréteur pragmatique du système de dialogue (cf. chapitre 1, mais aussi [35]) ajoute les énoncés humains π_{15} , π_{16} et π_{17} à la SDRS calculée à l'étape 1. ; ainsi, l'interpréteur calcule d'abord $Elaboration(\pi_{15}, \pi_{16})$, ensuite il calcule $QAP(\pi_{14}, Elaboration(\pi_{15}, \pi_{16}))$, $QAP(\pi_{14}, \pi_{17})$ et enfin, étant donné le contexte multi-locuteur, $MP(\pi_{16}, \pi_{17})$ ⁷ ;
3. La procédure détermine la relation MP calculée précédemment, donc elle commute en situation **B.** ; ensuite, la SDRS composée des énoncés π_{15} et π_{16} , avec la relation $Elaboration$ entre eux, et de l'énoncé π_{17} sont sélectionnés comme des points d'attachement *candidats* pour le tour courant de M ; par conséquent, la machine éclate d'abord son intention communicationnelle en trois énoncés π_{21} , π_{22} et π_{23} , pour ensuite calculer les relations $Elaboration(\pi_{21}, \pi_{22})$ et $Consequence(\pi_{22}, \pi_{23})$; enfin, π_{21} est joint à l'historique du dialogue *via* les relations $P-Elab(Elaboration(\pi_{15}, \pi_{16}), \pi_{21})$ et $P-Elab(\pi_{17}, \pi_{21})$;
4. L'interpréteur pragmatique ajoute d'abord π_{24} et π_{25} à l'historique du dialogue *via* les relations $IQAP(\pi_{23}, \pi_{24})$ et $IQAP(\pi_{23}, \pi_{25})$; ensuite, il détermine les relations $MP(\pi_{24}, \pi_{25})$ et $P-Corr(\pi_{24}, \pi_{25})$, donc, en écrivant de manière raccourcie, $MP \wedge P - Corr(\pi_{24}, \pi_{25})$;
5. La procédure trouve une relation MP entre les énoncés π_{24} et π_{25} , donc elle reste en situation **B.** ; puis, cette procédure détermine les points d'attachement π_{24} et π_{25} ; ensuite, le tour courant de M est structuré (du point de vue rhétorique), en calculant les relations rhétoriques monologiques $Contrast(\pi_{31}, \pi_{32})$ et $Consequence(\pi_{32}, \pi_{33})$; enfin, ce tour de parole est joint à l'historique du dialogue *via* les relations $P-Corr(\pi_{24}, \pi_{31})$ et $P-Corr(\pi_{25}, \pi_{31})$;
6. L'interpréteur pragmatique ajoute l'énoncé humain π_{34} à l'historique du dialogue, *via* les relations $Elab_q(\pi_{32}, \pi_{34})$ et $IQAP(\pi_{33}, \pi_{34})$;
7. La procédure de structuration rhétorique pour la génération⁸ (c'est-à-dire, les algorithmes décrits dans ce chapitre), en ne détectant pas de relations MP dans le tour de parole humain précédent, commute en situation **A.**, donc, en vertu de l'algorithme correspondant pour la mise à jour de la structure de discours, ajoute π_{41} à l'historique du dialogue, *via* la relation $QAP(\pi_{34}, \pi_{41})$;
8. L'interpréteur pragmatique calcule tout d'abord $MP(\pi_{42}, \pi_{43})$, car $\neg equals(emitte(\pi_{42}), emitte(\pi_{43}))$, ensuite $Elab_q(\pi_{42}, \pi_{43})$; enfin, cette sous-structure est jointe à l'historique du dialogue *via* $Background_q(\pi_{41}, \pi_{42})$ et $Elab_q(\pi_{41}, \pi_{43})$;
9. La structuration rhétorique, en détectant la relation MP calculée lors de l'étape précédente, commute en situation **B.** ; puis, elle éclate l'intention communicationnelle provenue du contrôleur de dialogue, en deux énoncés π_{51} et π_{52} , qu'elle relie, rhétoriquement, *via* la relation $Contrast(\pi_{51}, \pi_{52})$; ces énoncés sont ensuite joints à l'historique du dialogue *via* les relations $P-Corr(\pi_{43}, \pi_{51})$ et $QAP(\pi_{42}, \pi_{52})$;
10. L'interpréteur pragmatique calcule d'abord $MP(\pi_{53}, \pi_{54})$, ensuite $Elaboration(\pi_{54}, \pi_{55})$, pour enfin connecter ces énoncés à l'historique du dialogue, *via* les relations $ACK(\pi_{52}, \pi_{53})$ et $ACK(\pi_{52}, Elaboration(\pi_{54}, \pi_{55}))$;

⁷On suppose qu'à la différence des traitements décrits en [35], chapitre 5, pour l'interprétation pragmatique, l'interpréteur a été étendu afin qu'il intègre la relation MP .

⁸Nous allons appeler désormais cette procédure « structuration rhétorique ».

11. La structuration rhétorique détecte d'abord la relation MP calculée à l'étape précédente, par conséquent elle reste en situation **B.**, ensuite elle relie l'énoncé π_{61} (qui correspond, pour l'instant, à l'intention communicationnelle fournie sous forme logique par le contrôleur de dialogue) à l'historique du dialogue : $Elab_q(\pi_{53}, \pi_{61})$ et $Elab_q(Elaboration(\pi_{54}, \pi_{55}), \pi_{61})$;
12. L'interpréteur pragmatique relie d'abord π_{62} à l'historique du dialogue via $Q-Elab(\pi_{61}, \pi_{62})$, ensuite il calcule $MP(\pi_{62}, \pi_{63})$, car ces énoncés ont des émetteurs différents ; enfin, $P-Elab(\pi_{62}, \pi_{63})$ est aussi calculée ;
13. La structuration rhétorique décide d'abord de rester en situation **B.**, car une relation MP a été calculée par l'interpréteur pragmatique à l'étape précédente, ensuite elle éclate l'intention communicationnelle en deux énoncés, π_{71} et π_{72} , connectés via $Consequence(\pi_{71}, \pi_{72})$; enfin, cette sous-structure rhétorique est jointe à l'historique du dialogue via $QAP(\pi_{62}, \pi_{71})$ et $QAP(\pi_{63}, \pi_{71})$;
14. L'interpréteur pragmatique calcule d'abord deux relations MP entre les énoncés des locuteurs humains : $MP(\pi_{73}, \pi_{74})$ et $MP(\pi_{74}, \pi_{75})$; ensuite, il connecte ces énoncés à l'historique du dialogue, via les relations rhétoriques $ACK(\pi_{72}, \pi_{73})$, $ACK(\pi_{72}, \pi_{74})$ et $ACK(\pi_{72}, \pi_{75})$;
15. La structuration rhétorique détecte d'abord les deux relations MP calculées précédemment, par conséquent elle commute en situation **B.** et, en vertu de l'algorithme de mise à jour approprié, elle établit la liste des antécédents possibles dans l'historique du dialogue : les énoncés π_{73} , π_{74} et π_{75} ; ensuite, à un niveau monologique, la machine éclate son intention communicationnelle en trois énoncés π_{81} , π_{82} et π_{83} , connectés ainsi : $Contrast(\pi_{82}, \pi_{83})$ et $Elaboration(\pi_{81}, Contrast(\pi_{82}, \pi_{83}))$; enfin, cette sous-structure de discours est connectée aux antécédents possibles dans l'historique du dialogue, via les relations $P-Elab(\pi_{73}, \pi_{81})$, $P-Elab(\pi_{74}, \pi_{81})$ et $P-Elab(\pi_{75}, \pi_{81})$;
16. L'interpréteur pragmatique calcule d'abord deux relations $MP(\pi_{85}, \pi_{86})$ et $MP(\pi_{88}, \pi_{89})$, pour ensuite structurer les tours de parole de chaque locuteur humain : $Contrast(\pi_{84}, \pi_{85})$ pour L_3 , $Contrast(\pi_{86}, \pi_{87})$ et $Consequence(\pi_{87}, \pi_{88})$ pour L_1 et π_{89} pour L_2 ; ensuite, la relation dialogique $ACK(\pi_{88}, \pi_{89})$ est calculée ; enfin, cette sous-structure rhétorique est jointe à l'historique du dialogue via $QAP(Contrast(\pi_{82}, \pi_{83}), Contrast(\pi_{84}, \pi_{85}))$ et $QAP(Contrast(\pi_{82}, \pi_{83}), Contrast(\pi_{86}, \pi_{87}))$; nous notons par Π la structure rhétorique calculée à cette étape (et contenant les énoncés π_{84} à π_{89}) ;
17. La structuration rhétorique commute d'abord en situation **B.** (puisqu'elle détecte les relations MP calculées à l'étape précédente), ensuite elle éclate l'intention communicationnelle en deux énoncés π_{91} et π_{92} connectés via $Consequence(\pi_{91}, \pi_{92})$; enfin, elle joint cette structure à l'historique du dialogue via $P-Elab(\Pi, \pi_{91})$;
18. L'interpréteur pragmatique calcule d'abord deux relations MP : $MP(\pi_{93}, \pi_{94})$ et $MP(\pi_{94}, \pi_{95})$, ensuite il ajoute ces trois énoncés à l'historique du dialogue, via $QAP(\pi_{92}, \pi_{93})$, $QAP(\pi_{92}, \pi_{94})$ et $QAP(\pi_{92}, \pi_{95})$;
19. La structuration rhétorique commute d'abord en situation **B.** (car elle détecte les relations MP), ensuite elle éclate l'intention communicationnelle en deux énoncés, π_{101} et π_{102} ,

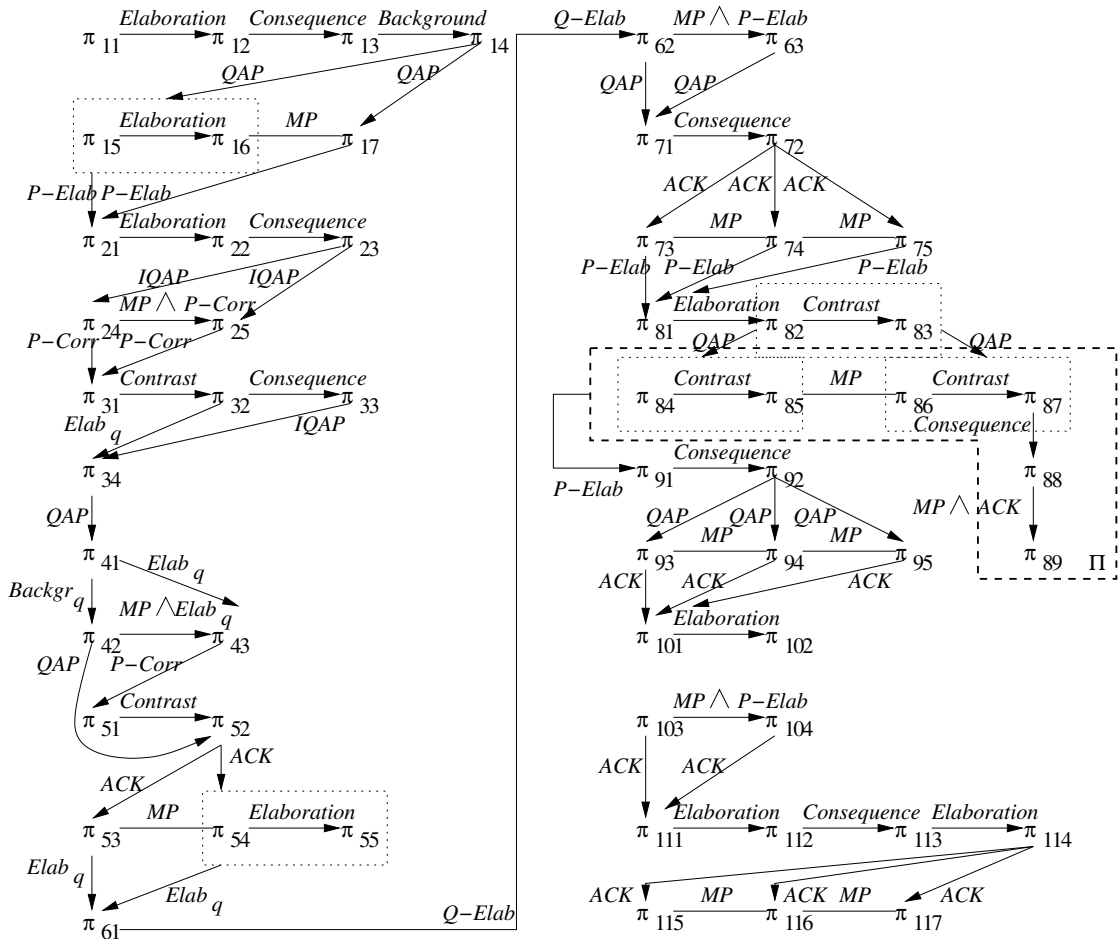


FIG. 5.4 – Structure rhétorique d'un dialogue à plusieurs locuteurs

qu'elle connecte *via* $Elaboration(\pi_{101}, \pi_{102})$; enfin, elle joint cette sous-structure à l'historique du dialogue, *via* $ACK(\pi_{93}, \pi_{101})$, $ACK(\pi_{94}, \pi_{101})$ et $ACK(\pi_{95}, \pi_{101})$;

20. L'interpréteur pragmatique calcule d'abord $MP(\pi_{103}, \pi_{104})$, pour ensuite connecter ces deux énoncés *via* $P-Elab(\pi_{103}, \pi_{104})$; enfin, il ajoute ces énoncés à l'historique du dialogue, *via* $QAP(\pi_{102}, \pi_{103})$ et $QAP(\pi_{102}, \pi_{104})$;
21. La structuration rhétorique commute d'abord en situation **B**. (car elle détecte la relation MP calculée à l'étape précédente), ensuite elle éclate l'intention communicationnelle en quatre énoncés, π_{111} , π_{112} , π_{113} et π_{114} , qu'elle connecte *via* les relations monologiques $Elaboration(\pi_{111}, \pi_{112})$, $Consequence(\pi_{112}, \pi_{113})$ et $Elaboration(\pi_{113}, \pi_{114})$; enfin, elle ajoute cette sous-structure rhétorique à l'historique du dialogue, *via* $ACK(\pi_{103}, \pi_{111})$ et $ACK(\pi_{104}, \pi_{111})$;
22. L'interpréteur pragmatique calcule d'abord les relations $MP(\pi_{115}, \pi_{116})$ et $MP(\pi_{116}, \pi_{117})$, ensuite il ajoute ces énoncés à l'historique du dialogue, *via* les relations $ACK(\pi_{114}, \pi_{115})$,

$ACK(\pi_{114}, \pi_{116})$ et $ACK(\pi_{114}, \pi_{117})$.

La structure rhétorique *globale* (c'est-à-dire, résultant en prenant en compte toutes les sous-structures rhétoriques, qui correspondent à tous les dialogues entre des paires d'interlocuteurs) calculée selon les étapes décrites ci-dessus est illustrée dans la figure 5.4, où nous avons encadré en polygones pointillés les constituants discursifs (soit étiquetés de manière distincte – comme Π , soit non-étiquetés de manière explicite).

5.5 Conclusions et perspectives

Dans ce chapitre nous avons proposé un cadre formel pour la structuration rhétorique dans *certaines* situations de dialogue multi-locuteurs. Ce cadre est appliqué à l'interaction homme-machine multi-locuteurs simultanés et illustré *via* plusieurs exemples. Ainsi, même si notre modélisation a, certes, des limites, telles que l'incapacité de gérer des dialogues multi-locuteurs généraux, où l'interaction n'est guidée par aucun « protocole » fixé d'avance, les algorithmes qui sont basés sur ce cadre permettent d'une part à une machine de gérer des interactions plus flexibles et complexes que les dialogues classiques bi-locuteurs, et établissent, d'autre part, une perspective formelle et systématique sur les dialogues à plusieurs locuteurs.

A long terme, le cadre décrit dans ce chapitre pourrait être à la base des applications plus complexes, impliquant en même temps des interactions entre des agents artificiels et des utilisateurs humains, dans des jeux sur ordinateur, ou dans des applications plus récentes de narration interactive [37], où un (ou plusieurs) sujet(s) humain(s) interagi(ssen)t de manière multimodale avec plusieurs agents artificiels. Ainsi, dans ce type d'applications, la manière dont les partenaires de l'interaction échangent des répliques est particulièrement pertinente pour une interaction naturelle et conviviale [171].

Enfin, pour les buts assumés dans notre thèse, ce cadre formel sera utilisé pour piloter des décisions concernant le réglage fin des énoncés produits par la machine en dialogue multi-locuteurs ; par conséquent, la structure rhétorique ainsi calculée sera utilisée pour contrôler plusieurs phénomènes linguistiques en génération, tels que les ellipses sémantiques (chapitre 6), le degré de force illocutoire (chapitre 7), ou les connecteurs entre des paires d'énoncés (chapitre 8).

Chapitre 6

Contrôle rhétorique de l'ellipse sémantique en génération pour le dialogue à plusieurs locuteurs

6.1 Introduction

Les chapitres 3, 4 et 5 portaient plutôt vers la structuration (rhétorique) des répliques dialogiques, tandis que ce chapitre (et ceux qui vont suivre) s'oriente vers la génération de surface, en montrant comment les contraintes rhétoriques agissent sur la forme des énoncés générés.

La gestion des ellipses en traitement automatique du langage naturel représente un sujet de recherche très important (tout particulièrement en génération, pour éviter les énoncés trop lourds et trop longs), surtout lorsqu'il s'agit du dialogue homme-machine, qui y pose des problèmes supplémentaires, tels que le caractère spontané et implicite des énoncés, ou bien le fort appui sur le contexte extra-linguistique (les déictiques, par exemple). L'ellipse peut être comprise à deux niveaux : d'abord, à un niveau profond (sémantique), on décide de ce qu'on peut éliminer (le *quoi* de la communication) ; ensuite, à un niveau superficiel (morpho-syntaxique), on décide de la manière dont on peut matérialiser, dans des énoncés, les décisions prises au niveau profond, le *comment* communiquer. Lorsque la génération du langage est concernée, cette distinction s'avère d'autant plus pertinente.

Dans le contexte des dialogues, soit entre un utilisateur et une machine, soit entre plusieurs utilisateurs avec intervention d'une machine, il importe que les réponses générées par la machine ne soient pas trop lourdes et fassent quelques ellipses à bon escient. Par exemple, si un locuteur cherche une tragédie sur un parricide, la machine, au lieu de répondre 'J'ai 13 tragédies antiques et deux modernes' peut répondre : 'J'ai 13 antiques et deux modernes', en réduisant ainsi la double reprise du mot 'tragédie' dont il vient d'être question ; si elle demande une précision sur l'auteur et que le locuteur répond 'Aristophane', alors la machine pourra répondre 'Il n'y a pas de tragédie d'Aristophane', évitant ainsi de répéter 'antique' après 'tragédie' en présupposant que le locuteur sait qu'Aristophane est un auteur antique. Cela n'est pas un problème simple à résoudre : il faut en effet utiliser la structure du discours pour savoir quelle information du

contexte (« accessible ») doit être restituée et au contraire ce qui peut être supprimé parce que la structure du discours permettra aisément à l'utilisateur de le récupérer.

Dans ce chapitre nous proposons un cadre formel pour contrôler, au niveau sémantique, les ellipses en génération du langage naturel pour le dialogue. D'abord, le cadre est particularisé pour le dialogue homme-machine traditionnel, comportant un seul locuteur humain ; ensuite, ces démarches sont étendues au dialogue multi-locuteurs ; il s'agit donc de proposer un cadre théorique et computationnel pour la prise des décisions sur la partie de l'intention communicationnelle qu'il est nécessaire et suffisant de mettre sous forme linguistique et communiquer aux allocutaires – c'est ce que nous appelons « ellipse sémantique ». Le traitement de l'ellipse se fait à deux niveaux : d'abord, au niveau sémantique, nous supprimons certaines parties de la forme logique d'un énoncé, pour réduire sa redondance tout en gardant sa pertinence par rapport au contexte ; ensuite, au niveau morpho-syntaxique nous ajustons la forme des énoncés afin que ceux-ci soient corrects (par exemple, si dans l'énoncé 'J'ai deux livres' nous supprimons le prédicat qui exprime l'objet de l'énoncé – object('book'), au niveau de surface il faut récupérer cet objet *via* une anaphore pronominale – 'J'*en* ai deux'). Seul le premier de ces deux niveaux est détaillé dans ce chapitre. De plus, nous n'avons pas considéré les ellipses conventionnelles (par exemple, 'Je vais à la poste' pour 'Je vais au bureau de poste', ou 'Je fais le plein (d'essence)'), ni les ellipses de proximité (répétitions).

Pour mieux clarifier ce que nous entendons par « ellipse » dans ce chapitre, nous précisons qu'il s'agit de l'élimination des prédicats dans les formes logiques des énoncés générés, non pas des conséquences linguistiques de ces suppressions. Il est vrai qu'au niveau linguistique une suppression d'un prédicat dans la forme sémantique peut se manifester soit sous forme de véritable ellipse linguistique (cf. le premier exemple montré ci-dessus), soit sous forme d'anaphore (cf. le deuxième exemple ci-dessus).

Pour d'autres travaux voisins, en ce qui concerne le dialogue traditionnel (à deux locuteurs), la théorie SDRT classique [11] offre des moyens pour gérer ces phénomènes linguistiques, surtout lorsqu'il s'agit de l'*interprétation* des discours ; par ailleurs, d'autres théories le font aussi, dans une certaine mesure, comme par exemple la RST (« Rhetorical Structure Theory ») [59]. Pour la génération, peu de travaux s'appuient sur la SDRT (voir par exemple [51]), sans plus parler du dialogue, où il n'y a presque rien sur la génération faisant appel à la SDRT. De plus, pour le dialogue à plusieurs locuteurs, il y a un certain vide au niveau théorique, vide que les recherches décrites dans ce chapitre tentent à combler. Ainsi, nous proposons ici un cadre pour contrôler, de manière guidée par des éléments de pragmatique, la génération des ellipses sémantiques en dialogue homme-machine à un ou plusieurs locuteurs humains. Pour cela, nous nous appuyons sur des connaissances sémantiques, rhétoriques et pragmatiques déterminées au préalable. Les formes logiques exprimant les énoncés à produire représentent le point de départ dans nos démarches, la structure rhétorique guide la prise de décision sur le maintien des prédicats dans la forme logique et, enfin, des connaissances pragmatiques sur la visée « affirmative » ou « contradictoire » des relations rhétoriques pour l'allocutaire raffinent davantage cette prise de décision.

L'idée de contrôler, au niveau sémantique, la génération des ellipses en dialogue homme-machine n'est pas nouvelle ; par exemple, les travaux de Jokinen [87] ont visé la planification « pragmatique » des contributions langagières appropriées en dialogue homme-machine s'ap-

puyant sur des idées dérivées des principes de communication coopérative de Grice¹ [11]. Ainsi, Jokinen fait une distinction entre les concepts implicites et explicites, l'ellipse étant liée au premier type précisé ; cet auteur n'utilise pas de structure rhétorique, ni de notion de cohérence du dialogue, mais s'appuie en revanche sur la notion de **pertinence** des énoncés par rapport aux attitudes des locuteurs, qu'elle quantifie à travers certains critères s'apparentant à la maxime de quantité de Grice [87].

Néanmoins, l'approche décrite dans ce chapitre se distingue des travaux de Jokinen, par l'appui sur la structure de discours formalisant la cohérence du dialogue, exprimée dans le cadre de la SDRT.

Le chapitre comprend, à la suite de l'introduction, trois paragraphes : le premier, présente de manière succincte le cadre choisi pour modéliser le dialogue entre la machine et plusieurs locuteurs, du point de vue de la génération ; le deuxième paragraphe décrit en détail les principes et idées de départ de notre modélisation lorsqu'il s'agit du dialogue traditionnel, comportant seulement deux locuteurs, ainsi que les algorithmes proposés pour cela ; l'approche est illustrée *via* un exemple tiré d'un corpus de dialogues ; les démarches décrites sont ensuite étendues au dialogue à plusieurs locuteurs et illustrées *via* un traçage sur un exemple détaillé ; une discussion sur certains aspects du cadre proposé achève ce paragraphe ; enfin, le dernier paragraphe conclut le chapitre et propose des pistes pour des recherches futures.

6.2 Situations de dialogue à plusieurs

Dans le chapitre 4 de cette thèse nous avons proposé un cadre formel adapté aux interactions du type dialogue entre plusieurs locuteurs ; comme nous avons déjà vu dans les chapitres 4 et 5, ce cadre peut être particularisé pour le cas où seule la perspective de la machine est adoptée, pour ensuite dégager deux situations conversationnelles typiques où la machine est censée s'engager dans un (ou des) dialogue(s) avec plusieurs locuteurs humains à la fois. Par conséquent, dans ce chapitre nous allons nous appuyer sur cette formalisation, dont nous reprenons ici seulement les éléments indispensables à la compréhension des développements proposés.

Pour cela, nous précisons d'abord un ensemble de notations utilisées :

- $L_1, \dots, L_N, L_\alpha, L_\beta, \dots ::=$ locuteur humain d'identité i , $i \in \{1, \dots, N\}$ ou α, β , etc. ;
- $M ::=$ la machine ;
- $SDRS_M^{\alpha\beta} ::=$ structure segmentée de discours (SDRS), exprimée dans le cadre de la théorie SDRT, pour le dialogue, vu par la machine, entre L_α et L_β ;
- $SDRS_{1NM} ::=$ réunion de toutes les SDRS vues (voire construites) par la machine à partir des dialogues entre les locuteurs L_i et L_j ainsi que $[i, j] \subseteq [1, N]$;
- $\pi(\alpha, \beta) ::=$ étiquette d'un énoncé provenant de L_α et destiné à L_β ;
- $\pi(\alpha, B) ::=$ étiquette d'un énoncé provenant de L_α et destiné aux locuteurs $L_\beta : \beta \in B$;
- $t(\pi) ::=$ indice ordinal de l'énoncé libellé π , dans la suite d'énoncés produits par l'émetteur de π dans le dialogue (à plusieurs) où ce locuteur est couramment engagé ;
- $emitter(\pi) ::=$ émetteur de l'énoncé libellé π ;
- $equals(n, m) ::=$ prédicat vrai si et seulement si. $n = m$ ou $n \equiv m$;

¹Cette approche est d'ailleurs choisie par beaucoup de chercheurs en génération du langage [147].

- Acte/1 ::= fonction précisant le type de l'acte de langage [185], [35] ;
- dest/1 ::= prédicat vrai si et seulement si le destinataire de l'énoncé dont la forme logique contient ce prédicat, coïncide avec l'argument de ce prédicat ;
- les symboles logiques $\exists, \forall, \Rightarrow, \neg, \wedge, \vee$ ont leur signification habituelle ;
- les symboles mathématiques $<, \leq, =, \geq, >, \cup, \cap, \emptyset, \subseteq, \subset, \supseteq, \supset, \in, \ni, \setminus$ ont leur signification habituelle.

Nous considérons deux situations de dialogue² du type service, comportant la machine et plusieurs (N) locuteurs humains :

- A. Dialogue du type service où la machine (par exemple en tant que bibliothécaire) parle simultanément à des clients qui ne parlent pas entre eux ; ce type de dialogue est aussi approprié aux interactions d'enseignement ou de communication publique, où les auditeurs parlent chacun à la machine, qui répond à son tour, compte tenu des connaissances du locuteur ayant le dernier, parlé à la machine, ou de plusieurs locuteurs (voire tous) ;
- B. Dialogue du type service (par exemple, bibliothèque virtuelle) ou d'enseignement³, où la machine est, respectivement, un client ou un élève qui ne parle pas à d'autres locuteurs humains ; ainsi, la machine dialogue avec un seul locuteur, mais elle tient compte de ce que celui-ci a parlé aux autres locuteurs humains. Pour illustrer la manière dont la machine peut intégrer des conversations en aparté des usagers, dans un dialogue à *but commun*, nous présentons ci-dessous un fragment d'un dialogue où deux locuteurs humains, L_i et L_j , sont des clients d'une bibliothèque (dont le bibliothécaire est la machine, M), où ils essaient d'emprunter des matériaux documentaires (les répliques sont désignées par des indices inférieurs pour la machine, et supérieurs pour les deux locuteurs) ; la machine écoute le dialogue s'engageant entre L_i et L_j et enfin elle propose une solution qui satisfait les souhaits des deux locuteurs :

M_1 : Alors, vous voulez une comédie française moderne, c'est ça ?

L_j^1 : Oui, un truc sur des vices liés à l'argent, par exemple un avare, ou quelque chose comme ça...

L_i^1 : Non, attends, il vaut mieux qu'on prenne des vices de gaspillage, par exemple le jeu ou quelque chose comme ça, n'est-ce pas ?

L_j^2 : Oui, ce serait marrant si on trouvait ce que tu dis, mais je doute qu'il y ait une telle comédie française sur cela, moi je n'en connais aucune...

L_i^2 : En fait, permets-moi de te contredire, il y en a au moins une, moi par exemple j'ai lu « Le joueur » de Regnard.

L_j^3 : Ah bon, et c'est quoi ça, une comédie de mœurs ?

L_i^3 : Oui, tout-à-fait ; en fait, il s'agit d'un gars qui joue même l'amour de sa copine au trictrac, j'ai trouvé la pièce bien sympa, j'aimerais bien la relire !

²Ces deux situations ont déjà été évoquées au chapitre 5.

³L'enseignant parle en général à un groupe, donc le destinataire de ses messages est un locuteur « moyen » ; dans ce cas, la machine pourrait se représenter un profil de ce locuteur moyen – par exemple, lorsqu'il s'agit d'un groupe d'élèves en quatrième année au lycée et l'enseignant présente des notions en calcul intégral, ce dernier peut supposer que ces élèves connaissent la notion de limite des fonctions sur la droite réelle. De toutes façons, la construction des profils de l'usager « moyen » n'est pas pour l'instant abordée dans nos recherches.

L_j^4 : OK, tu m'as convaincu, donc il faut voir s'ils l'ont à cette bibliothèque...

L_i^4 : Tu vas l'adorer, tu verras !

M_5 : En fait, voilà, nous avons cette pièce, parue en 1927 chez Hachette Classiques, je pourrai vous la prêter pour deux semaines ! Alors, je vous la prête ?

Il faut préciser que dans la modélisation présentée ici la machine ne fait pas *d'a priori* sur les connaissances des locuteurs ; celles-ci sont inférées seulement à partir des *engagements* publics des locuteurs. Ces engagements publics, représentés sous forme d'une liste pour chaque locuteur, constituent la base pour le calcul des *pires de prédicats* « connus » par chaque locuteur humain (voir les paragraphes 6.3.1 et 6.3.2). Cela suppose que le locuteur humain (l'usager) « sait » ce qu'il a affirmé lui-même, ou confirmé, à propos de ce qu'un autre locuteur (y compris la machine) avait dit. Ainsi, comme nous le verrons dans les paragraphes 6.3.2 et 6.3.2, la machine essaye, lorsqu'elle s'adresse à un groupe de locuteurs (même si cela se passe suite à une réplique en provenance d'un seul locuteur), de produire une réplique assez explicite pour qu'elle soit compréhensible pour *tous* les membres du groupe. En tout cas, au début du dialogue (quand les piles des usagers sont vides), la machine produit des énoncés aussi explicites que possible (ce qui revient à réaliser sous forme linguistique tous les prédicats dans la forme logique exprimant l'intention communicationnelle). Ce n'est donc qu'au fur et à mesure que le dialogue progresse, que la machine s'autorise à produire des énoncés elliptiques (donc, à supprimer des prédicats dans l'intention communicationnelle, selon les mécanismes présentés dans les paragraphes 6.3.1 et 6.3.2).

Dans le cas **A.**, le contexte discursif tel que vu par M est :

$$SDRS_{1NM} = \bigcup_{i=1}^N SDRS_M^{Mi}.$$

Pour la mise à jour de ces structures discursives lorsque la machine est en train de générer un énoncé, M reçoit (du contrôleur de dialogue [35]) une forme logique, correspondante à l'intention communicationnelle à mettre sous forme linguistique :

$$\tilde{K}(\pi(M, I)) = \text{Acte}(\bigwedge_{i \in I} \text{dest}(L_i) \wedge K(\pi(M, i))),$$

pour produire l'énoncé étiqueté π adressé aux locuteurs $\{L_i : i \in I\}$, avec $I \subseteq \{1, \dots, N\}$. Cette forme logique doit s'ajouter aux structures de discours $\{SDRS_M^{Mi} : i \in I\}$.

Dans le cas **B.**, le contexte discursif tel que vu par M est :

$$SDRS_{1NM} = SDRS_M^{Mi_0} \cup \left(\bigcup_{j=1; j \neq i}^N SDRS_M^{i_0j} \right),$$

où L_{i_0} est le locuteur « central », rendant le service (par exemple, le bibliothécaire ou l'enseignant).

Pour la mise à jour de ce contexte discursif, la machine doit produire sous forme linguistique une intention communicationnelle de la forme :

$$\tilde{K}(\pi(M, i_0)) = \text{Acte}(\text{dest}(L_{i_0}) \wedge K(\pi(M, i_0))).$$

Ici il y a un seul indice i_0 , correspondant au locuteur « central », L_{i_0} ; cette forme logique doit s'ajouter à la structure de discours $SDRS_M^{Mi_0}$, compte tenu des structures $\{SDRS_M^{i_0j} : j \in \{1, \dots, N\}; j \neq i_0\}$.

Ces deux situations sont suggérées dans la figure 6.1 ; dans le cas **A.**, la machine est toujours un locuteur « impliqué », c'est-à-dire, qui participe directement au dialogue, tandis que dans le cas **B.**, la machine est soit un locuteur « observateur », donc qui ne fait que suivre le dialogue (se déroulant entre deux locuteurs humains), soit un locuteur « impliqué » – lorsque la machine

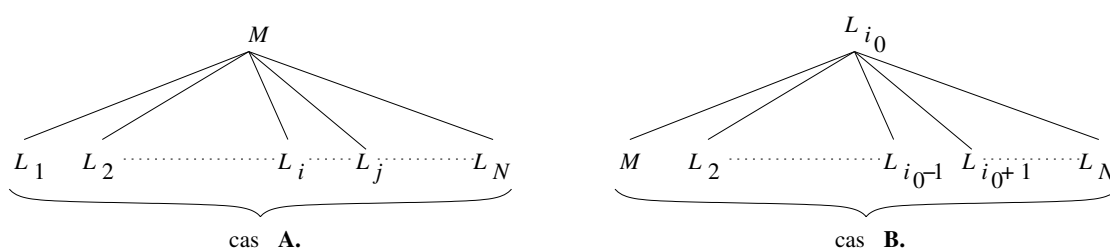


FIG. 6.1 – Situations de dialogue à plusieurs locuteurs

parle au locuteur « central ». Nous avons constaté, en étudiant des pièces de théâtre contenant des dialogues à plusieurs locuteurs, que la plupart de ces conversations peuvent être vues (et exprimées) comme alternances de situations **A.** et **B.**

6.3 Gestion de l'ellipse sémantique

6.3.1 Cas du dialogue traditionnel

Cadre théorique

Notre approche du contrôle sémantique des ellipses s'appuie à la fois sur la structure rhétorique (la SDRS) formalisant la cohérence du dialogue et sur les prédicats (dans une **ontologie de tâches** – voir chapitre 1 et [35]) exprimant des énoncés *affirmés*⁴ par l'utilisateur lors du dialogue, ou affirmés par la machine et *confirmés* par l'utilisateur.

En ce qui concerne les relations rhétoriques, nous utilisons le même sous-ensemble de 17 relations rhétoriques dans la SDRT, que nous avons déjà introduit dans le chapitre 3 ; ces relations sont regroupées, pour ce qui tient au contrôle des ellipses, en deux catégories :

- Relations rhétoriques *de confirmation* : à travers ces relations l'énoncé courant (apparaissant en tant que deuxième argument d'une relation rhétorique) ne met pas en question l'énoncé antérieur auquel il est relié ; ces relations rhétoriques, divisées à leur tour en relations *dialogiques* – reliant des énoncés en provenance des locuteurs différents et *monologiques* – en provenance du même locuteur, sont énumérées ci-dessous :
 - relations dialogiques : Question Elaboration (*Q-Elab*), Plan Elaboration (*P-Elab*), Elaboration_q (*Elab_q*), Narration_q, Question-Answer Pair (*QAP*), Acknowledgement (*ACK*) et Indirect QAP (*IQAP*) ;
 - relations monologiques : *Background*, *Consequence*, *Elaboration*, *Narration* et *Parallel* ;
- Relations rhétoriques *de contradiction* : à travers ces relations l'énoncé courant met en question ou même conteste l'énoncé antérieur auquel il est relié ; ces relations, divisées aussi en dialogiques et monologiques, sont :
 - relations dialogiques : Plan Correction (*P-Corr*), *Background_q*, et Not Enough Information (*NEI*) ;

⁴Les négations sont considérées ici des prédicats affirmés dans le sens que l'utilisateur s'engage pour ceux-ci.

– relations monologiques : *Alternation* et *Contrast*.

La modélisation des connaissances que l'utilisateur possède sur le contexte du dialogue et de la tâche se fait *via* une **pile** de prédicats (dans l'ontologie de tâches) exprimant au niveau sémantique les énoncés composant le dialogue. La structure de pile est motivée par le fait qu'il y a plus de chance que des références *implicites* à des concepts plus *récents* soient plus *compréhensibles* (pour l'allocutaire).

Ainsi, cette pile contient des prédicats (exprimant les énoncés dialogiques) :

- affirmés par l'utilisateur (y compris sous forme niée) ;
- affirmés par la machine et *confirmés* (de manière explicite ou implicite) par l'utilisateur.

La confirmation explicite des prédicats est faite *via* un énoncé relié par une relation rhétorique de confirmation à l'énoncé contenant ces prédicats, tandis que la confirmation implicite est faite *via manque* d'énoncé relié par une relation rhétorique de contradiction à l'énoncé contenant les prédicats en question.

On présente ci-dessous un exemple sur la manière dont la pile de prédicats est construite, en considérant le fragment de dialogue suivant (les indices supérieurs du type π_{ij} désignent les étiquettes des énoncés, i désigne l'indice du tour de parole courant et j désigne l'indice de l'énoncé en question, dans le cadre du tour de parole courant) :

M : Bienvenue sur Groplan ^{π_{11}} . Ce système vous permet de trouver un matériau bibliographique dans la bibliothèque de notre Université ^{π_{12}} . Vous pouvez à tout moment obtenir une aide en disant mode d'emploi général ^{π_{13}} . Que désirez-vous ? ^{π_{14}}

U : Bouquin de théâtre tragédie antique euh... théâtre grec ancien ^{π_{15}}

Le contenu sémantique de l'énoncé π_{15} produit par U , calculé par le module de compréhension dans l'architecture de dialogue [35], est exprimé dans une logique du premier ordre (voir les chapitres antérieurs) :

$$\exists X, Y, Z1, Z2, Z3, Z4 : \text{agent}(X) \wedge \text{equals}(X, \text{emitter}(\pi_{15})) \wedge \text{object}(Y) \wedge \text{equals}(Y, \text{'book'}) \wedge \text{feature}(Y, Z1) \wedge \text{equals}(Z1, \text{'theatre'}) \wedge \text{feature}(Y, Z2) \wedge \text{equals}(Z2, \text{'greek'}) \wedge \text{feature}(Y, Z3) \wedge \text{equals}(Z3, \text{'ancient'}) \wedge \text{feature}(Z3, Z4) \wedge \text{equals}(Z4, \text{'tragedy'}) \wedge \text{equals}(\Delta t(\pi_{15}), t).$$

La pile mise en discussion ci-dessus contient les prédicats *réifiés*, extraits de la forme logique montrée ci-dessus ; dans le contexte des recherches décrites dans ce chapitre, la réification consiste à éliminer les quantificateurs existentiels et les variables non-typées $X, Y, Z1, \dots, Z4$, en les reliant à leurs valeurs (précisées par le prédicat *equals/2*, éliminé à son tour). Ainsi, suite à la production de l'énoncé π_{15} , la machine ajoute une entrée dans la pile, sous forme logique conjonctive (n_k désigne le niveau numéro k , $k \geq 0$ dans la pile ; pour chaque énoncé dans le dialogue, un niveau est créé dans la pile) :

$$n_4 ::= \text{agent}(\text{emitter}(\pi_{15})) \wedge \text{object}(\text{'book'}) \wedge \text{feature}(\text{'book'}, \text{'theatre'}) \wedge \text{feature}(\text{'book'}, \text{'greek'}) \wedge \text{feature}(\text{'book'}, \text{'ancient'}) \wedge \text{feature}(\text{'ancient'}, \text{'tragedy'}) \wedge t(\pi_{15}).$$

Supposant que la pile de prédicats contient ceux auxquels l'utilisateur s'est **engagé** (dans le sens où il les a affirmés ou perçus et confirmés), le contrôle des ellipses sémantiques se réalise *via* un algorithme d'opération, spécifié ci-dessous. Pour préciser l'algorithme de manière formelle, nous utilisons les notations suivantes :

- $K(\pi) ::=$ formule logique exprimant le contenu sémantique de l'énoncé π ;
- $\Pi ::=$ pile de prédicats connus et non-infirmés par l'utilisateur ;

- \in ::= opération d'inclusion ponctuelle (d'un élément dans un ensemble) ;
- \notin ::= opération de non-inclusion ponctuelle (d'un élément dans un ensemble) ;
- $\rho^{(c)}$::= relation rhétorique de confirmation ;
- $\rho^{(\neg c)}$::= relation rhétorique de contradiction ;
- ρ_d ::= relation rhétorique dialogique ;
- ρ_m ::= relation rhétorique monologique ;
- \leftarrow ::= opération d'attribution de valeurs.

Ainsi, l'algorithme pour le contrôle des ellipses sémantiques en génération pour le dialogue homme-machine est le suivant :

pour un énoncé courant π :

pour chaque prédicat $p \in K(\pi)$:

1. **si** $p \in \Pi$ **et** :

- **s'il** existe un énoncé π_- antérieur à π **ainsi que** $p \in K(\pi_-)$ **et** il existe une relation rhétorique $\rho^{(c)}(\pi_-, \pi)$,
 - **ou s'il** existe encore un énoncé π_+ antérieur à π_- **ainsi que** $p \in K(\pi_+)$ **et** il existe des relations rhétoriques $\rho_d^{(\neg c)}(\pi_+, \pi_-)$ et $\rho_m^{(\neg c)}(\pi_-, \pi)$,
- alors** $K(\pi) \leftarrow K(\pi) \setminus \{p\}$;

2. **sinon**,

- **si** $p \notin \Pi$,
 - **ou si** $p \in \Pi$ **et** il existe un énoncé π_- antérieur à π **ainsi que** $p \in K(\pi_-)$ **et** il existe une relation rhétorique $\rho_d^{(\neg c)}(\pi_-, \pi)$,
- alors** garder p en $K(\pi)$.

De manière informelle, l'algorithme ci-dessus stipule que **si** l'énoncé courant contient un prédicat « de tâche » **et** ce prédicat se trouve dans la pile **et** ce prédicat apparaît dans un énoncé antérieur, relié à l'énoncé courant par une relation rhétorique dialogique non-contradictoire, **ou si** le prédicat est dans la pile, **mais** l'énoncé courant est relié par une relation rhétorique monologique de contradiction (par exemple, *Contrast*), suivant un énoncé relié par une relation dialogique de contradiction, **alors** nous supprimons ce prédicat dans l'énoncé courant ; **sinon, si** la relation rhétorique (cf. ci-dessus) est contradictoire (par exemple, *P-Corr*, *NEI*, etc.), **ou si** le prédicat n'apparaît pas dans la pile, **alors** nous gardons ce prédicat dans l'énoncé courant.

Pour la mise à jour de la pile de prédicats, il s'agit de prendre des décisions sur la présence (l'insertion) d'un prédicat de tâche dans la pile des prédicats (connus par l'usager) ; cela se fait *via* l'algorithme précisé de manière formelle ci-dessous :

pour un énoncé courant π :

pour chaque prédicat $p \in K(\pi)$:

1. **si** U est l'émetteur de π , **ou si** M est l'émetteur de π **et** il n'existe pas d'énoncé π_+ ultérieur à π (c'est-à-dire, appartenant au tour de parole *suivant*), émis par U **ainsi qu'il** existe $\rho_d^{(\neg c)}(\pi, \pi_+)$ **et** $p \in K(\pi_+)$,
alors $\Pi \leftarrow \Pi \cup \{p\}$;
2. **sinon, alors** ne pas insérer p dans Π .

De manière informelle, cet algorithme précise que **si** le prédicat de tâche est contenu dans un énoncé en provenance de l'utilisateur **ou si** le prédicat de tâche est contenu dans un énoncé en provenance de la machine, mais confirmé (de manière explicite ou implicite) par l'utilisateur, *via* un énoncé relié à l'énoncé mis en discussion précédemment par une relation rhétorique de « confirmation » (*ACK, QAP, P-Elab, Q-Elab*, etc.), **alors** nous insérons ce prédicat dans la pile ; **sinon**, nous n'insérons pas le prédicat dans la pile.

Exemple en dialogue homme-machine

Afin d'illustrer le mécanisme proposé pour le contrôle sémantique des ellipses en génération des réponses dialogiques de la machine, nous considérons un fragment de dialogue typique :

M_1 : Bienvenue sur Groplan ^{π_{11}} . Ce système vous permet de trouver un matériau bibliographique dans la bibliothèque de notre Université ^{π_{12}} . Vous pouvez à tout moment obtenir une aide en disant mode d'emploi général ^{π_{13}} . Que désirez-vous ? ^{π_{14}}

U_1 : Bouquin de théâtre tragédie antique euh... théâtre grec ancien ^{π_{15}}

M_2 : J'en ai trouvé 13 ^{π_{21}} . Vous pouvez donner plus de précisions, par exemple sur l'auteur, ou consulter les solutions ou effectuer une nouvelle recherche ^{π_{22}} .

U_2 : Aristophane ^{π_{23}} .

M_3 : Il n'y a pas de tragédie grecque ancienne par Aristophane ^{π_{31}} . Par contre, il y en a par les auteurs Sophocle, Euripide ou Eschyle ^{π_{32}} . Ces auteurs vous intéressent ? ^{π_{33}}

D'abord, face à un utilisateur la machine produit un tour de parole structuré en quatre énoncés (π_{11} à π_{14}), afin de déterminer une réponse de l'utilisateur ; celle-ci arrive, sous forme de l'énoncé π_{15} . Cela détermine la création de l'entrée n_4 dans la pile des prédicats Π , contenant la conjonction des prédicats réifiés, montrés dans le paragraphe 6.3.1. Ensuite, la machine produit la réplique M_2 contenant les énoncés π_{21} et π_{22} ; la sémantique de l'énoncé (en tant que fournie par le contrôleur de dialogue [35], mais voir aussi chapitre 1) π_{21} revient à la conjonction des prédicats réifiés :

$\text{agent}(\text{emitter}(\pi_{21})) \wedge \text{object}(\text{'book'}) \wedge \text{feature}(\text{'book'}, \text{'theatre'}) \wedge \text{feature}(\text{'book'}, \text{'greek'}) \wedge \text{feature}(\text{'book'}, \text{'ancien'}) \wedge \text{feature}(\text{'ancien'}, \text{'tragedy'}) \wedge \text{count}(\text{'book'}, 13) \wedge t_{-}(\pi_{21})$.

De plus, l'énoncé π_{21} est relié à l'énoncé antérieur π_{15} *via* la relation rhétorique de confirmation *P-Elab*(π_{15}, π_{21}), et les prédicats $\text{object}(\text{'book'})$, $\text{feature}(\text{'book'}, \text{'theatre'})$, $\text{feature}(\text{'book'}, \text{'greek'})$, $\text{feature}(\text{'book'}, \text{'ancien'})$ et $\text{feature}(\text{'ancien'}, \text{'tragedy'})$ font partie à la fois de l'énoncé π_{21} et de la pile des prédicats Π , donc, selon l'algorithme pour le contrôle sémantique des ellipses, tous ces prédicats sont supprimés de la forme logique de l'énoncé π_{21} . Ainsi, les prédicats réifiés exprimant la sémantique de l'énoncé π_{21} sont :

$\text{agent}(\text{emitter}(\pi_{21})) \wedge \text{count}(\text{'book'}, 13) \wedge t_{-}(\pi_{21})$.

Dans cette dernière forme logique, le premier argument du prédicat $\text{count}/2$ est une valeur liée à des prédicats éliminés de la forme logique, donc acquiert un marqueur de *non-réalisation* sous forme linguistique ; ainsi, ce prédicat devient $\text{count}(*, 13)$.

Ensuite, l'utilisateur produit l'énoncé π_{23} , ajoutant des précisions sur le livre souhaité, ce qui revient, au niveau logique, à l'ajout d'un niveau supplémentaire dans la pile, contenant le prédicat réifié $\text{author}(\text{'book'}, \text{'aristophanes'})$. L'énoncé π_{23} est relié par une relation rhétorique de confirmation à l'énoncé de la machine, π_{22} : *IQAP*(π_{22}, π_{23}).

Enfin, la machine produit son troisième tour de parole, contenant les énoncés π_{31} et π_{32} ; π_{31} est relié par une relation de contradiction à π_{23} : $P\text{-Corr}(\pi_{23}, \pi_{31})$. Même si une partie de ses prédicats réifiés se trouvent déjà dans la pile (il s'agit de $\text{object}(\text{'book'})$, $\text{feature}(\text{'book'}, \text{'theatre'})$, $\text{feature}(\text{'book'}, \text{'greek'})$, $\text{feature}(\text{'book'}, \text{'ancien'})$ et $\text{feature}(\text{'ancien'}, \text{'tragedy'})$), du fait que la relation rhétorique reliant π_{31} aux énoncés antérieurs ayant déterminé la mise à jour de la pile, est de contradiction, l'ellipse sémantique sur l'objet de l'énoncé n'est plus autorisée. Par contre, pour l'énoncé π_{32} , du fait qu'il est relié à π_{31} par une relation rhétorique monologique de contradiction ($\text{Contrast}(\pi_{32}, \pi_{31})$), l'ellipse sur l'objet de l'énoncé est autorisée, selon l'algorithme précisé.

Il est vrai que, dans la réplique M_3 , la machine pourrait supprimer à bon escient les déterminants 'grecque ancienne' du mot 'tragédie', en ne gardant que le déterminant 'Aristophane'. En termes sémantiques, la machine devrait supprimer dans la formule logique de cet énoncé les prédicats réifiés $\text{feature}(\text{'book'}, \text{'greek'})$ et $\text{feature}(\text{'book'}, \text{'ancien'})$. Nous croyons qu'une telle décision devrait s'appuyer sur le présupposé que U sait que Aristophane est un auteur grec ancien. Pourtant, l'équivalence entre 'théâtre grec ancien' et 'tragédie grecque ancienne' n'est pas calculée, ce qui fait que, même si les prédicats $\text{feature}/2$ précisant les déterminants 'grec ancien' attachés à l'entité 'théâtre' sont dans la pile de U , ils ne sont pas enlevés lorsqu'ils sont appliqués à l'entité 'tragédie' ; une telle suppression a besoin de l'information (présente dans l'ontologie de tâches) que les tragédies sont un genre particulier de théâtre. Pour l'instant, ce type de raisonnement « ontologique » n'est pas inclus dans l'algorithme.

6.3.2 Extensions au dialogue à plusieurs

Principes

Lors du passage au dialogue entre la machine et plusieurs locuteurs humains, les algorithmes pour la gestion de l'ellipse sémantique en génération des énoncés subissent quelques modifications, ainsi :

- la machine maintient un ensemble de piles de prédicats, une pile pour chaque locuteur humain ; ensuite, l'ensemble des prédicats à supprimer dans la forme logique correspondant à l'intention communicationnelle est calculé en considérant comme entrée dans l'algorithme pour la gestion de l'ellipse sémantique, l'*intersection* des contenus de toutes les piles de prédicats ;
- la mise à jour des piles de prédicats se fait de manière différente, compte tenu des confirmations **indirectes** des énoncés de la machine (voir ci-dessous).

Le premier volet ci-dessus exprime un choix de « bon sens », celui d'exprimer (sous forme linguistique) l'intention communicationnelle d'une manière aussi explicite que nécessaire pour qu'elle soit comprise par *tous* les usagers destinataires du message contenu dans cette intention. Cela revient à supprimer, dans l'intention communicationnelle, seulement les prédicats « connus » par tous les destinataires (c'est-à-dire, les prédicats qui se trouvent dans les piles de tous les destinataires), et à garder les autres.

En termes computationnels, cela revient, pour un ensemble de N piles Π_1 à Π_N à en choisir le sous-ensemble $\{\Pi_i : i \in I\}$, avec $I \subseteq \{1, \dots, N\}$ l'ensemble des destinataires du message à mettre sous forme linguistique par la machine, à calculer un ensemble Π_I que nous mettons en

entrée de l'algorithme pour la gestion des ellipses (tel que présenté dans le paragraphe 6.3.1) :

$$\Pi_I = \bigcap_{i \in I} \Pi_i.$$

Dans Π_I nous gardons l'ordonnancement des prédicats, en tant que précisé par chacune des piles individuelles Π_i .

Ensuite, il nous reste à préciser la manière dont les piles Π_1 à Π_N des usagers sont mises à jour. Dans le cas du dialogue à plusieurs, un prédicat p faisant partie d'une forme logique d'un énoncé d'étiquette π (soit produit par un des usagers, soit à produire par la machine) est ajouté dans la pile Π_i d'un locuteur quelconque L_i si et seulement si :

1. l'énoncé π est produit par le locuteur L_i -même ;
2. l'énoncé π est produit par la machine (M) et confirmé, de manière explicite ou implicite, par le locuteur L_i ; cette confirmation peut se faire de manière *directe* ou *indirecte*, comme nous présentons ci-dessous.

Donc, il y a essentiellement deux types de confirmations, explicites ou implicites, qui se classifient à leur tour en directes ou indirectes, comme nous montrons de manière synthétique ci-dessous (nous supposons que nous sommes en situation 2 ci-dessus, où l'énoncé π est produit par la machine) :

- la confirmation *explicite directe* a lieu lorsqu'il existe une relation rhétorique dialogique de confirmation $\rho_d^{(c)} \in SDRS_M^{Mi}$ reliant π à un énoncé π' ultérieur à π , ainsi que $\text{equals}(L_i, \text{emitter}(\pi'))$; nous allons considérer un exemple à cet égard, supposant que la machine entretient un dialogue avec deux locuteurs humains (L_i et L_j) autour d'une tâche concernant l'emprunt d'un livre à une bibliothèque publique (les indices numériques supérieurs désignent les tours de parole successifs – pour les usagers, tandis que pour la machine, ce sont les indices inférieurs qui désignent les tours de parole) :

M_1 : Bonjour, Messieurs ! Que pourrais-je faire pour vous ?

L_i^1 : Eh ben, je voudrais une pièce mythologique de Molière, si vous en avez !

L_j^1 : Et moi également !

M_2 : Eh ben, je pourrais vous donner « Amphytrion », est-ce que ça vous ira ?

L_i^2 : Oui, tout-à-fait !

- la confirmation *explicite indirecte* a lieu lorsqu'un autre locuteur L_j , $j \neq i$ produit un énoncé π' ultérieur à π ainsi qu'il y ait une des deux situations suivantes :

1. il existe une relation rhétorique dialogique de confirmation $\rho_d^{(c)}$ reliant π' à π **et** le locuteur L_i produit l'énoncé π'' ainsi qu'il existe une relation rhétorique dialogique de confirmation $\bar{\rho}_d^{(c)}$ reliant π'' à π' ; un fragment de dialogue, dans le même contexte que celui montré ci-dessus, est :

M_1 : Eh ben, je pourrais vous donner « Amphytrion », est-ce que ça vous ira ?

L_j^1 : Oui, « Amphytrion » est parfait pour moi !

L_i^1 : Pour moi aussi, je suis d'accord !

2. il existe une relation rhétorique dialogique de contradiction $\rho_d^{(-c)}$ reliant π' à π **et** le locuteur L_i produit l'énoncé π'' ainsi qu'il existe une relation rhétorique dialogique de contradiction $\bar{\rho}_d^{(-c)}$ reliant π'' à π' ; un exemple de dialogue est donné ci-dessous :

M_1 : Eh ben, je pourrais vous donner « Amphytrion », est-ce que ça vous ira ?

L_j^1 : Non, je n'aime pas cette pièce !

L_i^1 : Attendez, par contre pour moi ça m'ira très bien !

- la confirmation *implicite directe* a lieu lorsqu'après que la machine produit π , le locuteur L_i ne produit aucun énoncé relié de manière rhétorique à π et il existe un autre locuteur humain L_j , $j \neq i$, qui produit un autre énoncé π' ultérieur à π ainsi qu'il n'y ait aucune relation rhétorique entre π et π' ; un exemple à cet égard est donné ci-dessous :

M_1 : Eh ben, je pourrais vous donner « Amphytrion », est-ce que ça vous ira ?

L_j^1 : Attendez... qu'est-ce que vous pensez du film « La pianiste », avec Isabelle Huppert ? Moi, j'aimerais bien le voir !

Dans ce dialogue L_j , parlant à la machine, change directement de topique, lors du tour de parole L_j^1 ; toutefois, afin que le dialogue soit cohérent, L_j confirme implicitement (via une autre modalité non-linguistique, par exemple en remuant la tête de manière affirmative, juste au début de la production de L_j^1) à la machine l'énoncé M_1 . De toute manière, ces aspects non-linguistiques ne sont pas modélisés de façon explicite dans le cadre que nous proposons, mais celui-ci peut être étendu afin qu'il intègre les actes de dialogue de nature non-linguistique [35].

- la confirmation *implicite indirecte* a lieu lorsqu'après que la machine produit π , un autre locuteur L_j , $j \neq i$ produit un énoncé π' ainsi qu'il existe une relation rhétorique dialogique de confirmation $\rho_d^{(c)} \in SDRS_M^{Mj}$ reliant π' à π et L_i ne produit aucun énoncé relié de manière rhétorique à π' ou à π ; un fragment de dialogue illustrant cette situation est montré ci-dessous :

M_1 : Eh ben, je pourrais vous donner « Amphytrion », est-ce que ça vous ira ?

L_j^1 : Oui, ça m'ira à merveille !

L_i^1 : Qu'est-ce que vous pensez du film « La pianiste », avec Isabelle Huppert ? Moi, j'aimerais bien le voir !

Tous ces traitements se font au niveau de chaque tour de parole, ce qui veut dire que les confirmations sont attendues lors de la prochaine prise de parole des locuteurs concernés.

Il est toutefois possible qu'un prédicat déjà inclus dans la pile d'un certain usager, L_i , soit *infirmé* par celui-ci dans un moment ultérieur à l'insertion de ce prédicat dans la pile. Dans ce cas, à l'instar des confirmations, il peut y avoir plusieurs types d'infirmités : directes et indirectes ; de toute façon, nous posons qu'il n'y a pas d'infirmités implicites, ce qui revient au fait qu'un prédicat est enlevé de la pile d'un usager si et seulement si celui-ci l'infirmé de manière explicite. Même si cela paraît intuitif, pour des raisons de complétude et précision, nous allons détailler ci-dessous les situations d'infirmité (explicite) possibles⁵ (on suppose toujours qu'il s'agit d'un prédicat p compris dans la forme logique correspondante à l'énoncé π produit par la machine – M) :

- l'infirmité *directe* a lieu lorsque le locuteur L_i produit l'énoncé π' ultérieur à π , ainsi qu'il y ait au moins deux tours de parole de différence entre la production de π' et celle

⁵On suppose toujours que la pile d'un usager quelconque, L_i , est concernée.

de π , et qu'il existe une relation rhétorique dialogique de contradiction $\rho_d^{(-c)} \in SDRS_M^{Mi}$ reliant π' à π ; un exemple illustrant cette situation est montré ci-dessous :

M_1 : Eh ben, je pourrais vous donner « Amphytrion », est-ce que ça vous ira ?

L_j^1 : Oui, « Amphytrion » est parfait pour moi !

L_i^1 : Pour moi aussi, je suis d'accord !

M_2 : D'accord, eh ben alors je mets le livre sur votre carte ?

L_j^2 : Oui, tout-à-fait !

L_i^2 : Attendez, en fait je me suis trompé, j'ai déjà lu cette pièce, donc je voudrais bien autre chose !

– l'infirmité *indirecte* a lieu dans une des situations suivantes :

1. il y a un autre locuteur L_j , $j \neq i$ qui produit un énoncé π'' ainsi qu'il y ait au moins deux tours de parole de différence entre π'' et π , qu'il existe une relation rhétorique dialogique de contradiction $\bar{\rho}_d^{(-c)} \in SDRS_M^{Mj}$ entre π et π'' , et que L_i produit un énoncé π''' ultérieur à π'' ainsi qu'il existe une relation rhétorique dialogique de confirmation $\tilde{\rho}_d^{(c)} \in SDRS_M^{ij}$ reliant π''' à π'' ; exemple :

M_1 : Eh ben, je pourrais vous donner « Amphytrion », est-ce que ça vous ira ?

L_j^1 : Oui, « Amphytrion » est parfait pour moi !

L_i^1 : Pour moi aussi, je suis d'accord !

M_2 : D'accord, eh ben alors je mets le livre sur votre carte ?

L_j^2 : En fait, non, attendez, je viens de me rendre compte que nous avons cette pièce à la maison, donc du coup...

L_i^2 : Oui, effectivement, nous l'avons déjà... c'est vrai...

2. il y a un autre locuteur L_j , $j \neq i$ qui produit un énoncé π'' ainsi qu'il y ait au moins deux tours de parole de différence entre π'' et π , qu'il existe une relation rhétorique dialogique de confirmation $\bar{\rho}_d^{(c)} \in SDRS_M^{Mj}$, et que L_i produit un énoncé π''' ultérieur à π'' ainsi qu'il existe une relation rhétorique dialogique de contradiction $\tilde{\rho}_d^{(-c)} \in SDRS_M^{ij}$ reliant π''' à π'' ; exemple :

M_1 : Eh ben, je pourrais vous donner « Amphytrion », est-ce que ça vous ira ?

L_j^1 : Oui, « Amphytrion » est parfait pour moi !

L_i^1 : Pour moi aussi, je suis d'accord !

M_2 : D'accord, eh ben alors je mets le livre sur votre carte ?

L_j^2 : Oui, quant à moi c'est bon, vous pouvez la mettre sur ma carte.

L_i^2 : Non, attendez, sur ta carte (la carte de L_j – n.a.) t'as pris cinq livres déjà, donc je pense qu'il vaut mieux qu'on mette cette pièce sur ma carte !

Traçage sur une situation de dialogue à plusieurs

Dans ce paragraphe nous allons présenter un exemple étendu de dialogue à trois locuteurs humains, L_1 , L_2 et L_3 , auxquels s'ajoute la machine, M . Le dialogue concerne l'emprunt de matériaux documentaires dans une bibliothèque, où la machine est le bibliothécaire et les trois locuteurs humains, les clients. C'est M qui commence le dialogue, en se présentant. Le dialogue est annoté en termes de locuteurs, tours de parole et énoncés, de la manière suivante :

$$M_j : \langle \text{énoncé} \rangle^{\pi_{j_0}} \dots \langle \text{énoncé} \rangle^{\pi_{j_1}} ;$$

$$L_k^j : \langle \text{énoncé} \rangle^{\pi_{j(i+1)}} \dots \langle \text{énoncé} \rangle^{\pi_{j_2}} .$$

Ici, j est l'indice du tour de parole, k est l'indice du locuteur humain, et π_{j_s} désigne le s ème énoncé dans le j ème tour de parole. Ainsi, nous présentons ci-dessous la manière dont la machine calcule les ellipses sémantiques qu'elle pourrait autoriser, afin que ses énoncés soient à la fois naturels et pertinents pour les usagers concernés (c'est-à-dire, destinataires des énoncés). De plus, nous marquons l'effet linguistique des ellipses sémantiques par des tirets bas encadrés entre crochets et identifiés par une paire d'indices inférieurs droits, dont le premier désigne le tour de parole concerné et le deuxième, l'ellipse en discussion ; donc, l'effet linguistique de la même ellipse sémantique dans le n ème tour de parole est marqué par $[\]_{nm}$. Ici les deux locuteurs ont un but commun et le dialogue est distribué entre eux pour arriver à ce même but, afin que le traitement de l'ellipse sémantique soit visible au niveau des énoncés produits par la machine ; ces deux locuteurs forment donc un « groupe coopératif » :

M_1 : Bonjour, je suis Groplan ^{π_{11}} , je peux vous assister dans la recherche d'un document dans notre bibliothèque ^{π_{12}} . Vous pouvez à tout moment vous renseigner sur la disponibilité d'un certain livre, CD ou DVD ^{π_{13}} . Que voudriez-vous faire ? ^{π_{14}}

L_1^1 : Bonjour ^{π_{15}} , eh ben, nous voudrions une pièce de théâtre sur un inceste ^{π_{16}} ...

L_2^1 : Une tragédie antique ou un truc comme ça ^{π_{17}} ...

M_2 : Attendez que je regarde ^{π_{21}} ... j'ai trouvé trois tragédies antiques où un inceste apparaît ^{π_{22}} . Pourriez-vous apporter plus de précisions, par exemple sur l'auteur ? ^{π_{23}}

L_1^2 : Aristophane ? ^{π_{24}}

L_3^2 : Ou, sinon, Virgile ? ^{π_{25}}

M_3 : Il n'y a pas de tragédie antique écrite par Aristophane ou Virgile et comportant un inceste ^{π_{31}} . Par contre, j'[en]₃₁ ai trouvé trois par Sophocle, une [₃₂] par Eschile, une [₃₃] par Euripide ^{π_{32}} . Voudriez-vous [en]₃₄ choisir une, connaître d'autres auteurs [₃₅], ou bien effectuer une nouvelle recherche ? ^{π_{33}}

L_2^3 : Qu'en avez-vous par Sophocle ? ^{π_{34}}

M_4 : Eh ben, par exemple [₄₁] [₄₂] « Oedipe roi » ^{π_{41}} ...

L_1^4 : De quoi s'agit-il dans cette pièce ? ^{π_{42}}

L_3^4 : Y a-t-il un inceste entre frère et sœur ? ^{π_{43}}

M_5 : En fait, il n'y a pas d'inceste entre frère et sœur dans la tragédie grecque antique « Oedipe roi » ^{π_{51}} , mais [₅₁] il s'agit, en gros, d'un parricide, suivi par un inceste entre mère et fils ^{π_{52}} .

L_2^5 : Ah oui, ça a l'air bon ^{π_{53}} ...

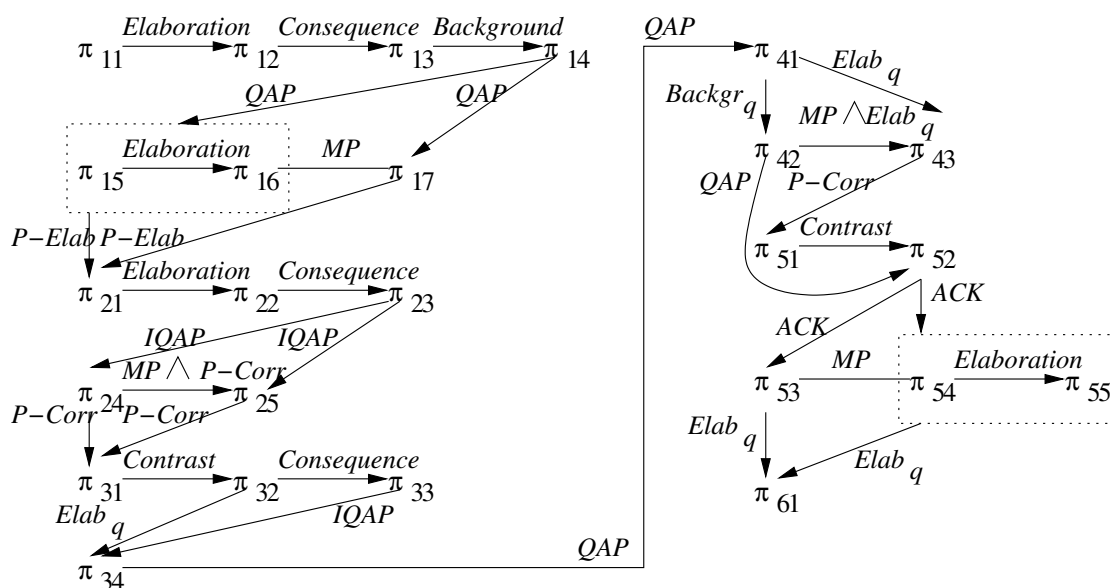


FIG. 6.2 – La structure rhétorique d'un dialogue à plusieurs locuteurs

L_1^5 : Alors, du coup c'est comme dans le film « Ma mère » avec Isabelle Huppert ^{π_{54}} .

J'ai adoré ce film ! ^{π_{55}}

M_6 : Alors, [ça]₆₁ vous convient ? ^{π_{61}}

D'abord, nous précisons que seules les ellipses sémantiques dans les énoncés produits par la machine sont concernées. Afin de montrer comment chaque ellipse est calculée, nous nous appuyons d'abord sur les structures de discours qui rendent compte de ce dialogue ; la manière dont cette structure de discours est calculée n'est pas présentée en détail ici.

Ainsi, nous présentons dans la figure 6.2 la réunion des structures de discours en discussion ; sur cette figure, l'étiquette *MP* désigne une « relation » de discours qui rend compte d'une situation de dialogue à plusieurs locuteurs ; cela veut dire que cette relation relie le dernier énoncé en provenance d'un locuteur au premier énoncé en provenance d'un autre locuteur (suivant), sauf la machine, dans un tour de parole donné.

D'abord, lors des premiers deux tours de parole, la machine n'autorise aucune ellipse sémantique, puisqu'il s'agit d'établir un arrière-plan commun, concernant les souhaits des utilisateurs (clients) et les disponibilités de la bibliothèque. Ensuite, à partir du troisième tour de parole, la machine autorise :

1. dans le troisième tour de parole :
 - (a) trois ellipses sémantiques, sur les résultats de la recherche, dans π_{32} ;
 - (b) une ellipse sémantique dans l'énoncé π_{33} ;
2. dans le quatrième tour de parole :
 - (a) deux ellipses sémantiques dans l'énoncé π_{41} ;
3. dans le cinquième tour de parole :

- (a) une ellipse sémantique dans l'énoncé π_{52} ;
- 4. dans le sixième tour de parole :
 - (a) une ellipse sémantique dans l'énoncé π_{61} .

Ci-dessous nous montrons en détail la manière dont chaque ellipse est autorisée ou bloquée, analysant chaque énoncé dans chaque tour de parole produit par la machine ; puisque les formes logiques des énoncés sont exprimées à l'instar de la situation de dialogue classique (comportant un seul locuteur), analysée en détail dans le paragraphe 6.3.1, ici nous n'allons pas montrer les formes logiques complètes de tous les énoncés ; seuls les prédicats dont la suppression est autorisée dans ces formes logiques seront précisés.

Ceci étant, le traçage du calcul des ellipses sémantiques en génération est présenté ci-dessous :

1. les énoncés π_{11} à π_{14} sont stéréotypiques, indépendants du contexte dialogique, puisqu'ils sont produits face à tout nouveau utilisateur ; dans ce cas, lorsqu'il s'agit d'une présentation du bibliothécaire virtuel, il n'a aucune ellipse sémantique calculée ; cependant, d'un point de vue rhétorique, les relations monologiques de confirmation *Elaboration*(π_{11}, π_{12}), *Consequence*(π_{12}, π_{13}) et *Background*(π_{13}, π_{14}) sont calculées, selon les mécanismes détaillés dans les chapitres 3 à 5 ;
2. lorsqu'il s'agit des énoncés π_{21} à π_{23} , apparemment, nous serions censés autoriser une ellipse sémantique sur les prédicats $\text{object}(Y) \wedge \text{equals}(Y, \text{'book'}) \wedge \dots$ dans la forme logique $K(\pi_{22})$, puisque π_{22} est relié par la relation monologique de confirmation *Elaboration* à l'énoncé π_{21} qui, à son tour, est relié par la relation dialogique de confirmation *P-Elab* à l'énoncé π_{17} et au constituant formé des énoncés π_{15} et π_{16} (à leur tour reliés par une relation monologique de confirmation, *Elaboration*) ; de toutes façons, la pile des prédicats de L_1 contient, après la production par celui-ci de l'énoncé π_{16} , les prédicats réifiés $\dots \wedge \text{feature}(\text{'book'}, \text{'theatrical_play'}) \wedge \text{feature}(\text{'book'}, \text{topic}(\text{'incest'}))$, tandis que la pile de L_2 contient, après la production de l'énoncé π_{17} , les prédicats réifiés $\dots \wedge \text{feature}(\text{'book'}, \text{'tragedy'}) \wedge \text{feature}(\text{'tragedy'}, \text{'ancient'})$; nous observons ainsi que l'intersection de ces deux piles est *vide*, c'est-à-dire, une unification des termes les composant est impossible ;
3. lors du troisième tour de parole, les choses sont plus nuancées :
 - (a) pour l'énoncé π_{31} l'élimination des prédicats réifiés $\text{feature}(\text{'book'}, \text{'tragedy'}) \wedge \text{feature}(\text{'tragedy'}, \text{'ancient'}) \wedge \text{feature}(\text{'book'}, \text{author}(\text{'aristophanes'})) \vee \text{feature}(\text{'book'}, \text{author}(\text{'virgil'}))$ n'est pas autorisée, puisque, d'un côté les prédicats sur les auteurs ne se trouvent pas dans les piles de tous les locuteurs ayant pris la parole précédemment et, de l'autre côté, même si les prédicats sur le type de l'ouvrage se trouvent maintenant dans les piles des usagers (du fait qu'il y avait une confirmation implicite directe de L_1 et L_3 à π_{22} , *via* les énoncés π_{24} et π_{25}), l'énoncé π_{31} est relié par une relation dialogique de contradiction, *P-Corr* aux énoncés π_{24} et π_{25} ;
 - (b) pour l'énoncé π_{32} , l'élimination des prédicats réifiés $\text{feature}(\text{'book'}, \text{'tragedy'}) \wedge \text{feature}(\text{'tragedy'}, \text{'ancient'}) \wedge \text{feature}(\text{'book'}, \text{topic}(\text{'incest'}))$ est autorisée, seul le

- prédicat sur l'auteur doit être gardé (puisque'il apporte un plus d'informations, non présentes dans les piles des prédicats des interlocuteurs) ; ainsi, nous autorisons trois ellipses, $[_]_{31}$, $[_]_{32}$ et $[_]_{33}$; au niveau linguistique, en français ces ellipses requièrent l'utilisation du pronom « en » ; ces ellipses sont autorisées conformément à l'algorithme précisé dans le paragraphe 6.3.1, puisque l'énoncé π_{32} est relié à π_{31} *via* une relation monologique de contradiction, *Contrast* ;
- (c) pour l'énoncé π_{33} , les mêmes arguments que ceux pour π_{32} justifient l'autorisation de la suppression des prédicats sur le type d'ouvrage, ce qui revient, au niveau linguistique, à la réalisation des ellipses $[_]_{34}$ et $[_]_{35}$; cela est vrai parce que l'énoncé π_{33} est relié à π_{32} *via* une relation rhétorique de confirmation, *Consequence* ;
4. lors du quatrième tour de parole, la machine autorise l'élimination des prédicats réifiés $\text{have}(M, \text{'book'}) \wedge \text{feature}(\text{'book'}, \text{'tragedy'}) \wedge \text{feature}(\text{'tragedy'}, \text{'ancient'}) \wedge \text{feature}(\text{'book'}, \text{author}(\text{'sophocles'}))$, puisque, d'un côté, ces prédicats se trouvent dans les piles des clients (suite au dialogue ayant déjà eu lieu – voir les arguments présentés ci-dessus) et, de l'autre côté π_{41} est relié à π_{34} *via* une relation rhétorique dialogique de confirmation, *QAP* ; de toutes façons, l'élimination du prédicat sur le titre de la tragédie, $\text{feature}(\text{'book'}, \text{title}(\text{'oedipus_rex'}))$ n'est pas autorisée, puisque'il est évident que celui-ci ne se trouve pas dans les piles des trois interlocuteurs humains ;
5. lors du cinquième tour de parole, les choses sont plus nuancées :
- (a) pour l'énoncé π_{51} , les prédicats sur le sujet de la tragédie ($\text{feature}(\text{'book'}, \text{'tragedy'}) \wedge \text{feature}(\text{'tragedy'}, \text{'ancient'}) \wedge \text{feature}(\text{'book'}, \text{title}(\text{'oedipus_rex'})) \wedge \text{feature}(\text{'book'}, \text{topic}(\text{'incest'})) \wedge \text{feature}(\text{'incest'}, \text{'brother-sister'})$) ne peuvent pas être supprimés, puisque cet énoncé est relié à π_{43} *via* une relation dialogique de contradiction, *P-Corr* et en π_{43} le prédicat sur le sujet de la tragédie est raffiné ;
- (b) pour l'énoncé π_{52} , l'élimination des prédicats réifiés $\text{feature}(\text{'book'}, \text{'tragedy'}) \wedge \text{feature}(\text{'tragedy'}, \text{'ancient'}) \wedge \text{feature}(\text{'book'}, \text{title}(\text{'oedipus_rex'}))$ est autorisée, selon l'algorithme décrit en 6.3.1, puisque cet énoncé est relié à π_{51} *via* une relation monologique de contradiction, *Contrast* ; cela revient, au niveau linguistique, à produire l'ellipse $[_]_{51}$;
6. pour le sixième tour de parole, la machine produit l'énoncé π_{61} , suite aux énoncés π_{53} et π_{54} en provenance de L_2 et, respectivement, L_1 , qui confirment le propos de la machine (donc, les information sur le livre concerné) ; ainsi, tous les prédicats spécifiant ce livre sont maintenant dans les piles de L_1 et L_2 par confirmation explicite directe, et de L_3 , par confirmation implicite directe ; de plus, l'énoncé π_{61} est relié à π_{53} et au constituant formé des énoncés π_{54} et π_{55} *via* une relation dialogique de confirmation, *Elab_q* ; ainsi, la machine est autorisée à supprimer tous les prédicats qui spécifient l'objet du désir des usagers ; cela revient, au niveau linguistique, à l'utilisation du pronom démonstratif « ça », marqué par l'ellipse $[_]_{61}$.

Il est utile de remarquer que, contrairement au dialogue à deux locuteurs analysé dans le paragraphe 6.3.1, où il aurait fallu supprimer le déterminant 'antique', ici il faut le laisser car on ne sait pas si le deuxième locuteur sait que Aristophane est un auteur antique (il s'agit du

principe de minimisation de la mutualisation des connaissances – autrement dit, on suppose que les interlocuteurs humains partagent aussi peu de connaissances communes que possible).

Nous précisons que les mécanismes de réalisation au niveau *syntactique* des ellipses sémantiques ne sont pas concernés par ce chapitre ; cette couche de réalisation de surface peut être simulée par un ensemble de patrons syntaxiques (ou même expressions figées), pour chaque situation d'éllision sémantique ; cela est utile notamment lorsque nous nous posons le problème des implémentations pratiques, où les contraintes de temps réel sont importantes, dans des applications telles que les services d'assistance, les jeux sur ordinateur ou même les narrations interactives.

6.3.3 Discussion

Appui sur la structure de discours

La méthode décrite dans ce chapitre pour contrôler les ellipses sémantiques en génération pour le dialogue homme-machine à plusieurs locuteurs repose sur la structure rhétorique du dialogue, considérée comme calculée (acquise). Toutefois, déterminer l'ensemble des relations rhétoriques qui relient les tours de parole n'est pas une chose triviale et beaucoup d'études lui sont consacrées, surtout quand il s'agit d'*interpréter* les dialogues. De plus, pour la génération, une partie des études se sont concentrées sur les situations monologiques (par exemple, génération des bulletins météo en langue naturelle, à partir d'un ensemble d'informations disponibles dans une base de données [147]).

Lorsque la génération en contexte dialogique est concernée, nous avons essayé de combler le manque relatif d'études à cet égard en proposant un cadre formel dérivé de la SDRT, pour représenter et mettre à jour la structure discursive du dialogue, lorsqu'un énoncé est disponible sous forme logique et doit être produit sous forme linguistique par la machine. Ainsi, après avoir émulé un fragment de la SDRT [35] dans une logique du premier ordre (voir chapitre 3), nous avons proposé un ensemble d'algorithmes pour mettre à jour la SDRS dialogique (voir chapitres 3 et 4). Essentiellement, ces procédures reviennent à relier l'énoncé courant (à être généré par la machine) aux énoncés antérieurs en dialogue, en vérifiant si chaque paire d'énoncés (dont un, l'énoncé courant) satisfait la sémantique d'une des relations rhétoriques possibles. Pour cela, nous avons également proposé un ensemble de sémantiques formelles (exprimées dans une logique du premier ordre) pour chacune des 17 relations rhétoriques considérées pertinentes pour le dialogue (voir chapitre 3) ; ces expressions logiques suivent de près les sémantiques informelles des relations rhétoriques, proposées dans la SDRT classique [11]. Ensuite, afin d'optimiser la mise à jour de la structure rhétorique du point de vue computationnel et que les représentations obtenues soient en accord avec les intuitions humaines, nous avons opéré un filtrage des relations rhétoriques possibles entre des paires d'énoncés, selon les actes de langage réalisés lors de la production de ces énoncés (voir la deuxième partie du chapitre 3). Par exemple, entre une paire d'énoncés (appartenant à deux locuteurs différents) qui réalisent les actes de langage « faire-faire-savoir » (F^{FS}) et « faire-savoir » (F^S) [35] on ne peut avoir qu'un ensemble limité de relations rhétoriques possibles : *QAP*, *IQAP* et *P-Elab*. Ainsi, entre un énoncé produit par l'utilisateur et réalisant un F^{FS} « Où puis-je trouver le livre 'X' ? » et un énoncé produit par la machine et réalisant un F^S il peut y avoir les trois relations rhétoriques, selon la sémantique de

l'énoncé produit par la machine : *QAP* si la machine produit 'Il se trouve au bout de ce couloir', *IQAP* si la machine produit 'Juste à côté de la porte de sortie, à gauche', et *P-Elab* si la machine produit 'Le plan des rangées de livres se trouve dans le hall en entrée'.

Limites apparentes

En ce qui concerne les limites potentielles de la méthode proposée ici pour générer des constructions elliptiques, un désavantage apparent de l'algorithme réside dans la possibilité de produire des énoncés ambigus, voire non-interprétables. Par exemple, apparemment nous pourrions avoir des situations comme ci-dessous, où nous désignons par *U* un locuteur humain et par *M*, la machine :

U : J'ai un livre de tragédie ancienne.

M⁽¹⁾ : Moi comédie.

Ici, le tour dialogique *M*⁽¹⁾ aurait pu être réalisé suite à l'application de l'algorithme pour la gestion des ellipses ; sans appliquer cet algorithme, nous aurions obtenu l'énoncé *M*⁽⁰⁾ :

M⁽⁰⁾ : Moi, j'ai un livre de comédie.

Le « moi » apparaît puisque, d'un côté, dans l'énoncé produit par la machine, l'agent est l'émetteur de cet énoncé, dans l'énoncé produit par l'usager, l'agent est aussi l'émetteur *et* ces deux énoncés sont reliés par le même type de relation rhétorique (par exemple, *QAP*) à un hypothétique énoncé d'autrui, qui demandait, par exemple, à chacun des locuteurs quels livres ils avaient à offrir ou à recommander. De toute manière, suite à l'application de l'algorithme proposé dans ce chapitre, on n'obtient ni *M*⁽⁰⁾, ni *M*⁽¹⁾, pour les raisons qui vont être précisées ci-après. La formule logique pour l'énoncé 'Moi, j'ai un livre de comédie' est : $\text{agent}(\text{emitter}(\pi)) \wedge \text{object}(\text{'book'}) \wedge \text{feature}(\text{'book'}, \text{'comedy'}) \wedge \text{have}(\text{emitter}(\pi), \text{'book'})$; ici, π est l'étiquette de l'énoncé produit par la machine. Suite à l'application de l'algorithme proposé dans ce chapitre, nous retrouvons les prédicats $\text{object}(\text{'book'})$, $\text{feature}(\text{'book'}, \text{'comedy'})$ et $\text{have}(\text{emitter}(\pi), \text{'book'})$, car ils ont été « produits » par *U*. D'où, apparemment, l'énoncé *M*⁽¹⁾ est produit. En réalité, même si ces prédicats sont supprimés dans la formule logique de l'énoncé à être produit par la machine, nous remarquons que 'comedy' est une « feature » de l'objet, qui est le même que dans l'énoncé antérieur. Donc, « comédie » doit rester une caractéristique d'un objet (du même objet qu'auparavant – un livre), ce qui est marqué au niveau linguistique par un énoncé du type :

M⁽²⁾ : Moi, un de comédie.

Ici, donc, même si nous supprimons les prédicats, la contrainte que ce qui reste soit une caractéristique d'un objet est marquée en gardant l'article indéfini et la préposition (« de »).

De l'autre côté, si nous appliquons *ad litteram* l'algorithme, nous ne devons supprimer que les prédicats qui apparaissent sous la même forme dans tous les énoncés concernés. Cela revient, pour l'exemple considéré ici, à ne supprimer que le prédicat $\text{object}(\text{'book'})$. Ensuite, le prédicat $\text{feature}(\text{'book'}, \text{'comedy'})$ devient $\text{feature}(*, \text{'comedy'})$, pour marquer l'élimination de l'objet de cet énoncé. Pour le reste, la forme logique concernée ne change pas, ce qui revient à produire l'énoncé :

$M^{(3)}$: Moi, j'en ai un, de comédie.

Donc, dans ce cas, nous gardons l'agent et le verbe, mais nous marquons l'ellipse de l'objet par une construction anaphorique (le pronom « en »). Le genre correct de l'article indéfini (« un ») est récupéré ici des prédicats $\text{have}(\text{'book'}) \wedge \text{feature}(*, \text{'comedy'})$, où l'objet « book » se réalise en français par le mot « livre », dont le genre est connu, *via* une grammaire ou un dictionnaire⁶.

6.4 Conclusions et perspectives

Dans ce chapitre nous avons présenté et illustré un mécanisme de gestion pragmatico-sémantique des ellipses en génération pour le dialogue homme-machine ; pour les aspects pragmatiques, nous avons proposé un regroupement des relations rhétoriques en de « contradiction » et de « confirmation » par rapport aux locuteurs ; pour la sémantique, nous nous appuyons sur les formes logiques des énoncés et sur la structure de discours, déduite à partir de celles-ci (voir chapitres 3, 4 et 5).

Nous avons ensuite étendu au dialogue à plusieurs locuteurs ce cadre formel et computationnel, l'illustrant *via* plusieurs exemples de conversations entre la machine et un ensemble de locuteurs, autour d'une tâche concernant l'emprunt de livres à une bibliothèque publique. Nous avons ainsi observé que le passage du dialogue entre deux locuteurs aux conversations comportant plusieurs interlocuteurs n'est pas trivial lorsqu'il s'agit de piloter le comportement de la machine, afin qu'elle soit capable d'entretenir un dialogue naturel face aux humains. Dans ce cadre, nous avons observé que l'ellipse sémantique est un élément important qui permet d'assurer un degré de verbosité optimal aux énoncés produits par l'ordinateur, compte tenu du contexte interactionnel qui implique plusieurs sujets interagissant de manière simultanée.

Concernant les perspectives de ce travail, nous envisageons plusieurs voies de développement : (i) réaliser aussi une gestion au niveau superficiel (pour le français) des ellipses en génération pour le dialogue, (ii) renforcer le rôle des connaissances (représentées dans l'ontologie de tâches) dans le traitement des ellipses sémantiques, suivant une approche s'apparentant aux recherches de McShane [115], mais opérant des extensions et adaptations pour le dialogue homme-machine d'une part, et pour la génération d'une autre part.

Nous croyons que les extensions au dialogue à plusieurs locuteurs sont les plus fructueuses, parce qu'elles permettent d'envisager des applications améliorées en narration interactive ou jeux sur ordinateur ; ainsi, nous envisageons d'étendre le cadre proposé ici au delà du domaine purement langagier, donc de plonger ces mécanismes dans le multimodal, pour piloter les gestes.

⁶Ici, « * » agit donc comme un pointeur vers l'objet supprimé, qui apparaît toutefois en tant que deuxième argument du prédicat *have*.

Chapitre 7

Réglage fin des énoncés en génération : contrôle du degré de force illocutoire

7.1 Introduction

Les recherches décrites dans ce chapitre poursuivent les démarches présentées dans les chapitres précédents qui sont maintenant mises en valeur lors de la production des réponses. Ainsi, il s'agit de régler le degré de force illocutoire [185] pour les énoncés générés par la machine, en dialogue à plusieurs locuteurs, en prenant en compte des aspects pragmatiques, la structure rhétorique du dialogue, et les engagements publics des interlocuteurs.

Afin de contrôler le degré de force illocutoire, nous nous appuyons sur plusieurs idées :

1. L'utilisation des travaux de Muller, Maudet et Prévot concernant la combinaison entre les relations rhétoriques de la SDRT et les engagements publics des locuteurs [93] ; en fait, nous utilisons un mécanisme pour calculer les engagements publics à partir des relations rhétoriques, qui est similaire à celui de Muller *et al.* [111], [112] ;
2. L'utilisation des concepts de Vanderveken pour définir et formaliser le degré de force illocutoire [185], en l'adaptant à la génération, comme l'ont montré des études antérieures de Caelen [81], en utilisant une échelle numérique à cinq niveaux pour quantifier le degré de force pour chaque acte de langage.

La contribution principale de ce chapitre réside dans la combinaison des points 1 et 2 précisés ci-dessus. Pour ce faire, nous partons d'un ensemble de prémisses : tout d'abord, nous supposons que chacun des partenaires de dialogue (plus précisément, la machine M et l'ensemble $U_i : i \in I$ de $|I|$ locuteurs humains – I est l'ensemble des locuteurs, et $|I|$ est son nombre d'éléments) a une **liste d'engagements** qui contient les formules logiques des énoncés auxquels chaque locuteur s'engage ouvertement pendant la conversation [93] ; nous notons ces listes d'engagements par CS_M pour la machine et par CS_i pour ses interlocuteurs, U_i ; ensuite, nous stipulons que, du point de vue de la génération des réponses, le but de la machine est d'atteindre le succès du *dialogue*, et non pas obligatoirement le succès de la *tâche* (ce dernier est la responsabilité des contrôleurs de dialogue et de tâche dans l'architecture de dialogue assumée dans cette thèse –

voir chapitre 1, cf. aussi [35]); du point de vue des engagements des locuteurs, le succès du dialogue est atteint si et seulement si $CS_{U_i} \equiv CS_{U_j} \equiv CS_M \forall i, j \in I$. Autrement dit, plus les listes d'engagements des interlocuteurs sont logiquement équivalents, plus le dialogue a du succès. Enfin, à la différence de Maudet *et al.* [112] qui considèrent le point de vue d'un observateur *extérieur* du dialogue, nous adoptons le point de vue de la machine, ce qui implique le fait que le rôle du degré de force illocutoire dans le succès du dialogue peut être précisé comme suit :

$$\forall i \in I : CS_i \neq CS_M \Leftrightarrow \exists \lambda : (\lambda \in CS_i \wedge \lambda \notin CS_M) \vee (\lambda \in CS_M \wedge \lambda \notin CS_i),$$

donc, de manière informelle, les listes d'engagements d'un usager U_i et de M sont différentes si et seulement s'il existe une formule logique λ telle que soit λ est dans la liste d'engagements de M et non pas dans la liste d'engagements de U_i , soit inversement. Dans ce cas, M doit ajuster le degré de la force illocutoire¹ réalisée par λ , ainsi que soit U_i s'engage au contenu de λ , soit M concède de ne plus s'engager à cet énoncé (ou inversement).

Dans le même contexte, nous pouvons mentionner le modèle de dialogue de Traum *et al.* [176], implémenté avec des agents conversationnels qui interagissent dans un contexte de dialogue à plusieurs locuteurs (virtuels et humains) pour des applications diverses, allant de l'entraînement des troupes militaires, jusqu'à la narration interactive. Cette approche s'articule autour de deux aspects différents de l'interaction : (i) la *discursivité* des contributions langagières des agents – Traum *et al.* ne modélisent pas de manière explicite les contraintes rhétoriques imposées par l'historique du dialogue sur les tours de parole, et de ce fait leur approche n'arrive qu'à *simuler* des capacités discursives ; (ii) l'*intentionnalité* des contributions dialogiques – l'approche mentionnée arrive, en effet, à « moduler » la forme linguistique de surface pour les intentions communicationnelles, mais cela en passant par les *attitudes émotionnelles* des interlocuteurs, ce qui requiert l'accès à leur état intentionnel, ce qui peut se faire de manière assez fiable dans des contextes limités (par une tâche donnée), mais n'est pas générique.

Un autre travail à signaler est le système POLLY de Gupta *et al.* [71] : les auteurs proposent une architecture de système qui contrôle la forme de surface des énoncés en dialogue entre le système et un locuteur humain, à partir de la Théorie de la politesse de Brown et Levinson [28]. Gupta *et al.* s'appuient sur des éléments pragmatiques pour choisir une forme de surface adéquate pour un énoncé spécifié sous forme profonde (comme un plan, c'est-à-dire, une suite de préconditions et d'actions à réaliser si les préconditions sont satisfaites). Les auteurs se basent sur les contraintes imposées par les *menaces* d'un locuteur, ce qui revient essentiellement à prendre en compte trois facteurs contextuels : (i) le pouvoir que l'allocutaire a sur le locuteur, (ii) la distance sociale entre le locuteur et l'allocutaire et (iii) un niveau d'imposition de l'acte de langage réalisé dans l'énoncé. A partir de la Théorie de la politesse de Brown et Levinson, Gupta *et al.* proposent un contrôle assez fin de la forme de surface des énoncés, selon le type de menace que le locuteur subit ; le passage de ces consignes de politesse (appelées « stratégies » dans la Théorie de la politesse) aux formes linguistiques est réalisé de manière plutôt *ad-hoc*, car ce n'est qu'au niveau de la planification de l'énoncé dans la génération linguistique, qu'on colle des items lexicaux à la forme prototypique, neutre, de l'énoncé. A la différence de POLLY, nous proposons de prendre en compte d'autres facettes du contexte dialogique (c'est-à-dire, la

¹On rappelle qu'à l'instar de la terminologie de Caelen [35] ou de Vanderveken [185] pour nous les notions de *type d'acte de langage* (selon la taxinomie précisée dans le chapitre 3 de cette thèse) et de *force illocutoire* sont synonymes.

structure rhétorique du dialogue et les engagements publics des interlocuteurs), pour gérer essentiellement le même phénomène – le choix d’une forme linguistique adéquate, entre plusieurs alternatives pour exprimer le même contenu sémantique.

Ce chapitre est structuré comme suit : dans le paragraphe suivant nous précisons les détails formels concernant les idées de départ ; le troisième paragraphe présente de manière détaillée les mécanismes pour ajuster le degré de force illocutoire pour les énoncés produits par la machine, dans le cas particulier du dialogue entre la machine et un seul locuteur humain ; ensuite, dans le quatrième paragraphe ce cadre est étendu au cas du dialogue à plusieurs locuteurs ; l’efficacité de l’approche est démontrée *via* des exemples typiques dans une tâche de réservation de livres dans une bibliothèque. Enfin, un ensemble de conclusions et de perspectives clôt le chapitre.

7.2 Cadre général

7.2.1 Structure rhétorique et engagements publics

L’idée principale concernant les deux cadres formels sur la structure du dialogue que nous avons utilisés jusqu’ici (structure rhétorique et engagements) est que les relations rhétoriques (dans le cadre de la SDRT) déclenchent la mise à jour des listes d’engagements pour les locuteurs ; étant donné que les détails formels concernant l’approche initiale de Muller *et al.* à cet égard sont présentés en [111] et [112] entre autres, nous n’allons présenter ci-dessous que de manière synthétique les idées de base derrière cette approche, reformulées dans notre propre cadre formel. Pour ce faire, nous introduisons un ensemble de notations (en reprenant une partie des notations introduites dans les chapitres antérieurs) :

- U, M ::= partenaires de dialogue, un utilisateur et, respectivement, la machine ;
- L ::= un locuteur générique ; dans notre cadre, il s’agit soit de M , soit d’un locuteur humain U ;
- CS_L ::= liste d’engagements pour le locuteur L ; une telle liste d’engagements contient essentiellement deux choses : (i) les formules logiques qui expriment les sémantiques des énoncés auxquels le locuteur concerné s’est engagé (de manière explicite ou implicite – voir plus loin), et (ii) les formules logiques qui expriment les sémantiques des relations rhétoriques qui relient les énoncés présents dans la liste d’engagements ;
- π_i ::= étiquettes pour les énoncés, pour un entier positif i ;
- ρ_m ::= étiquettes pour les relations rhétoriques monologiques (voir chapitres 3 et 6 dans cette thèse) ; toute telle étiquette peut être une des suivantes : *Alternation*, *Background*, *Consequence*, *Elaboration*, *Narration*, *Contrast* et *Parallel* (voir chapitre 3) ;
- $\rho_d^{(c)}$::= étiquettes pour les relations rhétoriques dialogiques de **confirmation** (voir chapitre 6) ; toute telle étiquette peut être une des suivantes : *Q-Elab*, *IQAP*, *P-Elab*, *Elab_q*, *Narration_q*, *QAP*, et *ACK* ;
- $\rho_d^{(-c)}$::= étiquettes pour les relations rhétoriques dialogiques de **contradiction** (voir chapitre 6) ; toute telle étiquette peut être une des suivantes : *P-Corr*, *Background_q*, et *NEI* ;
- $K(\pi)$::= sémantique de l’énoncé π (exprimée dans une logique du premier ordre – voir chapitre 3 dans cette thèse) ;

- $\Sigma_\rho ::=$ sémantique de la relation rhétorique ρ , exprimée dans la même logique du premier ordre que les énoncés, mais en utilisant un ensemble de prédicats discursifs indépendants de la tâche – voir également chapitre 3 ;
- $\text{emitter}(\pi) ::=$ fonction discursive retournant le locuteur qui a produit l'énoncé π ;
- $\text{equals}(\alpha, \beta) ::=$ prédicat discursif binaire qui est vrai si la variable α est égale à la valeur β ;
- $\leftarrow_\pi ::=$ opération qui précise que la partie gauche est mise à jour avec la partie droite, *via* l'énoncé π ;
- $\alpha/\beta ::=$ le symbole $/$ précise que la proposition α est vraie seulement pourvu que la proposition β soit vraie également.

Ainsi, la mise à jour des listes d'engagements des locuteurs en dialogue² peut être décrite comme suit :

1. Cas des relations rhétoriques *monologiques* :

$$\forall L \in \{U, M\} : (\text{equals}(L, \text{emitter}(\pi_i)) \Rightarrow CS_L \leftarrow_{\pi_i} CS_L \cup \{K(\pi_i)\}) \wedge (\text{equals}(L, \text{emitter}(\pi_j)) \wedge \rho_m(\pi_i, \pi_j) \Rightarrow CS_L \leftarrow_{\pi_j} CS_L \cup \{K(\pi_j), \Sigma_{\rho_m(\pi_i, \pi_j)}\}) .$$

De manière informelle, cette règle précise que pour tout locuteur L (c'est-à-dire, soit un utilisateur U , soit la machine M) qui a produit les énoncés π_i et π_j , reliés *via* la relation rhétorique monologique ρ_m , sa liste d'engagements CS_L est mise à jour dans deux étapes, d'abord en ajoutant la sémantique de π_i (retournée par la fonction $K(\pi_i)$) après avoir produit π_i , ensuite en ajoutant la sémantique $K(\pi_j)$ de π_j et la sémantique de la relation rhétorique ρ_m (c'est-à-dire, $\Sigma_{\rho_m(\pi_i, \pi_j)}$) après avoir produit π_j .

2. Cas des relations rhétoriques *dialogiques* :

$$\begin{aligned} &\forall L_1, L_2 \in \{U, M\} : \neg \text{equals}(L_1, L_2) \wedge \text{equals}(L_1, \text{emitter}(\pi_i)) \wedge \text{equals}(L_2, \\ &\text{emitter}(\pi_j)) \wedge ((\rho_d^{(c)}(\pi_i, \pi_j) \Rightarrow (CS_{L_1} \leftarrow_{\pi_i} CS_{L_1} \cup \{K(\pi_i)\}) \wedge (CS_{L_2} \leftarrow_{\pi_j} CS_{L_2} \cup \\ &\{K(\pi_i), K(\pi_j), \Sigma_{\rho_d^{(c)}(\pi_i, \pi_j)}\})) \vee (\rho_d^{(-c)}(\pi_i, \pi_j) \Rightarrow (CS_{L_1} \leftarrow_{\pi_i} CS_{L_1} \cup \{K(\pi_i)\}) \wedge (CS_{L_2} \leftarrow_{\pi_j} \\ &CS_{L_2} \cup \{K(\pi_j), \neg K(\pi_i), \Sigma_{\rho_d^{(-c)}(\pi_i, \pi_j)}\}))) \wedge ((CS_{L_1} \leftarrow_{\pi_j} CS_{L_1} \cup \{K(\pi_j), \\ &\Sigma_{\rho_d^{(c)}(\pi_j, \pi_k)}\})/\exists \pi_k : \text{equals}(L_1, \text{emitter}(\pi_k)) \wedge \exists \rho_d^{(c)}(\pi_j, \pi_k) \wedge \rho_d^{(c)}(\pi_i, \pi_j)) \\ &\vee (CS_{L_1} \leftarrow_{\pi_j} CS_{L_1} \setminus \{K(\pi_i)\} \cup \{K(\pi_j), \Sigma_{\rho_d^{(c)}(\pi_j, \pi_k)}\})/\exists \pi_k : \text{equals}(L_1, \text{emitter}(\pi_k)) \wedge \\ &\exists \rho_d^{(c)}(\pi_j, \pi_k) \wedge \rho_d^{(-c)}(\pi_i, \pi_j) \vee (CS_{L_1} \leftarrow_{\pi_j} CS_{L_1} \cup \{\neg K(\pi_j), K(\pi_k), \Sigma_{\rho_d^{(-c)}(\pi_j, \pi_k)}\})/\exists \pi_k : \\ &\text{equals}(L_1, \text{emitter}(\pi_k)) \wedge \exists \rho_d^{(-c)}(\pi_j, \pi_k)) . \end{aligned}$$

De manière informelle, cette règle précise que pour deux partenaires de dialogue différents, un utilisateur U et la machine M , ainsi qu'un a produit l'énoncé π_i et l'autre, l'énoncé π_j , nous avons une des situations possibles suivantes :

- (a) une relation rhétorique dialogique de confirmation $\rho_d^{(c)}$ relie les énoncés π_i et π_j ; dans ce cas, la liste d'engagements pour le locuteur qui a produit π_i est mise à jour avec la sémantique de cet énoncé, tandis que l'émetteur de π_j met à jour sa liste d'engagements avec les sémantiques de π_i , π_j et de la relation rhétorique qui les connecte ;

²Tout au long de ce paragraphe nous considérons le cas du dialogue entre la machine et un utilisateur humain.

- (b) une relation rhétorique dialogique de contradiction $\rho_d^{(-c)}$ relie les énoncés π_i et π_j ; dans ce cas, la liste d'engagements pour le locuteur qui a produit π_i est mise à jour comme dans la situation précédente, tandis que l'émetteur de π_j met à jour sa liste d'engagements avec la sémantique de π_j , la sémantique *niée* de π_i (c'est-à-dire, $\neg K(\pi_i)$), et avec la sémantique $\Sigma_{\rho_d^{(-c)}(\pi_i, \pi_j)}$ de la relation rhétorique qui connecte ces deux énoncés.

Pour voir comment la liste d'engagements de l'émetteur de π_i est mise à jour après que l'autre locuteur a produit π_j , nous devons attendre qu'un autre énoncé, π_k , soit produit par l'émetteur de π_i ; alors, selon le type de la relation rhétorique dialogique qui relie π_k à π_j , et le type de la relation rhétorique qui relie les énoncés π_i et π_j , nous avons une des situations possibles suivantes :

- (a) la liste d'engagements du locuteur qui a énoncé π_i (et, ensuite, π_k) est mise à jour avec la sémantique de l'énoncé π_j *pourvu qu'*une relation rhétorique dialogique de confirmation $\rho_d^{(c)}$ relie l'énoncé π_k à π_j *et qu'*une relation rhétorique dialogique de confirmation ait été trouvée entre les énoncés π_i et π_j aussi ;
- (b) la liste d'engagements de l'émetteur de π_i (et, ensuite, de π_k) est mise à jour en *enlevant* la sémantique de π_i et en *ajoutant* la sémantique de π_j , *pourvu qu'*une relation rhétorique dialogique de confirmation $\rho_d^{(c)}$ relie l'énoncé π_k à π_j *et qu'*une relation rhétorique dialogique de contradiction $\rho_d^{(-c)}$ ait été trouvée entre les énoncés π_i et π_j ;
- (c) la liste d'engagements du locuteur qui a produit π_i (et, par la suite, π_k) est mise à jour avec la sémantique *niée* de l'énoncé π_j (c'est-à-dire, avec $\neg K(\pi_j)$), *pourvu qu'*une relation rhétorique dialogique de contradiction $\rho_d^{(-c)}$ relie l'énoncé π_k à π_j .

Bien évidemment, chaque fois qu'un locuteur produit un énoncé relié à des énoncés antérieurs *via* des relations rhétoriques, les sémantiques de ces relations rhétoriques sont ajoutées à la liste d'engagements de ce locuteur [112]. Cependant, dans le traitement formel qui suit, nous allons parfois ignorer l'ajout de ces sémantiques aux listes d'engagements des locuteurs, en ne retenant que les sémantiques (éventuellement niées) des énoncés.

7.2.2 Degré de force illocutoire

La formalisation de la force illocutoire, aussi que la représentation de son degré de force sont, d'une certaine manière, tributaires à des travaux antérieurs de Vanderveken [185] et de Caelen [81]. Ainsi, en reprenant une partie de la discussion menée dans les chapitres 3 et 6, nous supposons que l'entrée dans le module de génération est une expression logique de la forme $F(p)$, où

- F est une **force illocutoire** dans la terminologie de Vanderveken, ou un type d'acte de langage, dans la terminologie de Caelen : F^F (« faire-faire »), F^S (« faire-savoir »), F^{FS} (« faire-faire-savoir »), F^P (« faire pouvoir »), et F^D (« faire-devoir ») ;
- p est un **contenu propositionnel** dans les terminologies de Vanderveken et Caelen ; dans le cadre formel proposé ici, ce contenu propositionnel est assimilé à la formule logique

qui exprime la sémantique de l'énoncé qui réalise l'acte F ; un tel contenu propositionnel est noté par $K(\pi)$ (voir ci-dessus), si l'étiquette de l'énoncé $F(p)$ est π .

A l'instar des travaux de Vanderveken et Caelen, nous supposons que chaque acte de langage F peut être réalisé (par un énoncé produit par un interlocuteur) avec un certain degré de puissance ; nous appelons ce degré **degré de force illocutoire** et nous le désignons formellement par $\partial\phi$. Ainsi, la correspondance entre un énoncé, son contenu propositionnel sa force illocutoire et le degré de cette force peut être vue comme une suite de compositions fonctionnelles :

$$\pi \xrightarrow{K} K(\pi) \xrightarrow{F} F(K(\pi)) \xrightarrow{\partial\phi} \partial\phi(F(K(\pi)) \times K(\pi)),$$

ce qui pourrait aussi s'écrire comme $(\partial\phi \circ (F \circ K) \times K)(\pi)$. Nous présentons ci-dessous de manière explicite chacune de ces fonctions :

- La fonction K attribue des sens (exprimés sous la forme d'une formule logique du premier ordre) aux énoncés ou aux relations rhétoriques ; ainsi, elle met en correspondance des étiquettes d'énoncé avec des formules logiques :

$$K : \Pi \cup P \rightarrow \Lambda,$$

où $\Pi = \{\pi_1, \dots\}$ et $P = \{\rho_1, \dots\}$ sont deux ensembles d'étiquettes – pour les énoncés et pour les relations rhétoriques, $\Lambda = \{\lambda_1, \dots\}$ est un ensemble de formules logiques (exprimées en utilisant des prédicats dans une ontologie de tâches, pour les énoncés, ou dans l'ensemble des prédicats discursifs, pour les relations rhétoriques – voir chapitre 3 pour ces prédicats discursifs).

Ainsi, la fonction K est calculée, lorsqu'il s'agit de générer les énoncés de la machine, par le contrôleur de dialogue et, lorsqu'il s'agit d'analyser les énoncés des interlocuteurs, par le module d'analyse sémantique (cf. chapitre 1). Néanmoins, nous supposons que les représentations au niveau de l'analyseur sémantique et du contrôleur de dialogue sont cohérentes, au sens où le même ensemble de prédicats (dans l'ontologie de tâches) est utilisé. Lorsque la fonction K reçoit comme argument l'étiquette d'une relation rhétorique, son opération revient à une simple recherche de la sémantique de la relation rhétorique concernée (pour chaque type de relation rhétorique il y a une seule forme logique qui exprime sa sémantique, cf. chapitre 3).

- La fonction F attribue des types d'actes de langage aux (sémantiques des) énoncés ; cette fonction n'est pas définie sur les sémantiques des relations rhétoriques :

$$F : \Lambda \setminus \{\Sigma_\rho : \rho \in P\} \cup \{F^F, F^S, F^{FS}, F^P, F^D\};$$

ici, $\{\Sigma_\rho : \rho \in P\}$ représente l'ensemble des sémantiques de toutes les relations rhétoriques connues (dans notre cadre, il s'agit de 17 relations – voir chapitre 3) ; si la fonction F est appliquée sur une formule $K(\pi)$ telle que c'est la machine qui a produit l'énoncé π (donc $\text{equals}(M, \text{emitter}(\pi))$), alors le calcul de F est la responsabilité du contrôleur de dialogue.

- La fonction $\partial\phi$ attribue des degrés de puissance aux actes de langage qui sont à leur tour appliqués aux formes logiques des énoncés ; ainsi, $\partial\phi$ établit une correspondance entre les actes de langage et des nombres entiers.

Afin d'assurer plus de variabilité pour les énoncés produits par la machine, nous stipulons que la fonction $\partial\phi$ peut avoir, pour le même type d'acte de langage, des effets qui dépendent du contenu propositionnel de l'énoncé, $K(\pi)$. Ainsi, nous définissons la fonction $\partial\phi$ sur le produit cartésien entre l'ensemble des types d'actes et l'ensemble des contenus

propositionnels :

$$\partial\phi : \{F\} \times (\Lambda \setminus \{\Sigma_\rho : \rho \in P\}) \rightarrow \{-2, -1, 0, 1, 2\},$$

où $\{F\}$ et Λ désignent, respectivement, l'ensemble des types d'actes de langage possibles, et l'ensemble (potentiellement infini, mais limité en pratique par la tâche particulière considérée) des formules logiques générables à partir d'un ensemble précisé (par exemple, dans l'ontologie de tâches) de prédicats. A l'instar de [81], nous attribuons cinq niveaux possibles au degré de force illocutoire, en les classifiant en **très fort**, **fort**, **neutre**, **faible** et **très faible** ; -2 et -1 représentent les forces très faible et faible, 0 représente la force neutre, et 1 et 2 représentent les forces forte et très forte, respectivement.

Le but de ce chapitre est de définir de manière « analytique » la fonction $\partial\phi$.

Avant de faire cela, nous allons d'abord motiver cette démarche, en montrant, *via* un exemple, qu'il y a une relation pertinente entre le degré de force illocutoire attribué à un énoncé, et la forme linguistique de cet énoncé. Par exemple, supposons un dialogue entre un utilisateur U et une machine M , où l'utilisateur dit :

U : Excusez-moi, pourriez-vous me dire, s'il vous plaît, où je peux trouver le livre 'B' ?

Le contrôleur de dialogue en M produit, en tant que réponse à la question de U , une intention communicationnelle d'informer l'utilisateur que le livre se trouve au premier étage, à gauche ; cette intention communicationnelle est exprimée, sous forme logique, et offerte au générateur, comme nous le montrons ci-dessous (π désigne l'étiquette de cet énoncé) :

$$F^S(\exists X, Y, Z, T, U, V : \text{agent}(X) \wedge \text{equals}(X, \text{-emitter}(\pi)) \wedge \text{object}(Y) \wedge \text{equals}(Y, \text{'book'}) \wedge \text{feature}(Y, Z) \wedge \text{equals}(Z, \text{'title'}) \wedge \text{equals}(Z, \text{'B'}) \wedge \text{feature}(Y, T) \wedge \text{equals}(T, \text{'location'}) \wedge \text{feature}(T, U) \wedge \text{equals}(U, \text{'level'}) \wedge \text{equals}(U, \text{'1'}) \wedge \text{feature}(T, V) \wedge \text{equals}(V, \text{'direction'}) \wedge \text{equals}(V, \text{'left'})) .$$

Quelques formes linguistiques possibles, pour cette intention communicationnelle, selon les différents degrés pour force illocutoire F^S , sont (les marqueurs linguistiques associés aux degrés de force illocutoire sont soulignés) :

– $\partial\phi(F^S(K(\pi)), K(\pi)) = -2 \Rightarrow :$

M : Au premier étage, à gauche, je crois.

– $\partial\phi(F^S(K(\pi)), K(\pi)) = -1 \Rightarrow :$

M : Attendez que je regarde... voilà : le livre 'B' est au premier étage, à gauche.

– $\partial\phi(F^S(K(\pi)), K(\pi)) = 1 \Rightarrow :$

M : Vous pouvez certainement trouver le livre 'B' au premier étage, juste à gauche.

– $\partial\phi(F^S(K(\pi)), K(\pi)) = 2 \Rightarrow :$

M : Ecoutez vous pouvez certainement trouver le livre 'B' au premier étage, juste à gauche, vous comprenez ?

Comme nous le verrons dans le paragraphe suivant, du fait qu'il s'agit d'un acte F^S (d'informer), toutes ces formes résultent d'un degré de force illocutoire neutre qui est également possible :

$$\partial\phi(F^S(K(\pi)), K(\pi)) = 0 \Rightarrow :$$

M : Vous pouvez trouver le livre 'B' au premier étage, à gauche.

Après avoir vu cet exemple, nous observons vite que, du point de vue d'un générateur de réponses, il y a deux problèmes à résoudre concernant le degré de force illocutoire : (i) le calcul du degré de force illocutoire approprié, pour un énoncé donné, spécifié sous forme logique, et (ii) la correspondance entre un degré de force illocutoire précisé pour un énoncé donné, et une réalisation linguistique appropriée. Le premier problème constitue l'objet de ce chapitre ; le deuxième est résolu simplement en utilisant un ensemble d'expressions linguistiques figées à trous, pour chaque intention communicationnelle possible ; chaque expression figée est annotée par son degré de force illocutoire. Il s'agit alors de choisir l'expression figée correspondante, étant donné un degré de force précisé pour une forme logique à mettre sous forme linguistique.

7.2.3 Situations de dialogue à plusieurs locuteurs

Dans ce paragraphe nous allons donner un bref aperçu du cadre formel que nous proposons pour modéliser les aspects rhétoriques concernant la génération des réponses en dialogue à plusieurs locuteurs. La présentation détaillée de ce cadre fait l'objet du chapitre 5 ; par conséquent, nous allons nous contenter de ne rappeler ici que les aspects qui sont essentiels pour comprendre le mécanisme que nous proposons pour calculer le degré de force illocutoire en dialogue à plusieurs locuteurs. Ainsi, nous introduisons encore un ensemble de notations pour spécifier ce cadre :

- $N ::=$ le nombre des locuteurs humains en dialogue (pour faire le parallèle avec les notations introduites précédemment, nous avons que $N = |I|$) ;
- $L_i ::=$ le i ème locuteur humain, pour i de 1 à N ;
- $SDRS_M^{M\alpha} ::=$ la structure rhétorique calculée par la machine pour le dialogue entre elle-même (M) et le locuteur L_α ;
- $SDRS_M^{\alpha\beta} ::=$ la structure rhétorique calculée par la machine pour le dialogue entre les locuteurs L_α et L_β ;
- $SDRS_{1NM} ::=$ la structure rhétorique calculée par la machine sur les dialogues impliquant les locuteurs L_1 à L_N .

Dans notre cadre nous prenons en compte deux situations « élémentaires » de dialogue homme-machine à plusieurs locuteurs :

- A.** La machine M est un « serveur » (par exemple, un bibliothécaire virtuel) qui interagit avec un ensemble de N locuteurs humains, désignés par L_1, \dots, L_N , qui, à leur tour, n'interagissent pas entre eux ;
- B.** La machine M est un « client » d'un serveur, instancié cette fois-ci par un locuteur humain, désigné par L_{i_0} ; ainsi, M n'interagit qu'avec L_{i_0} , tandis que ce dernier interagit aussi avec les autres locuteurs humains qui, eux, n'interagissent pas entre eux (cf. le cas **A.**).

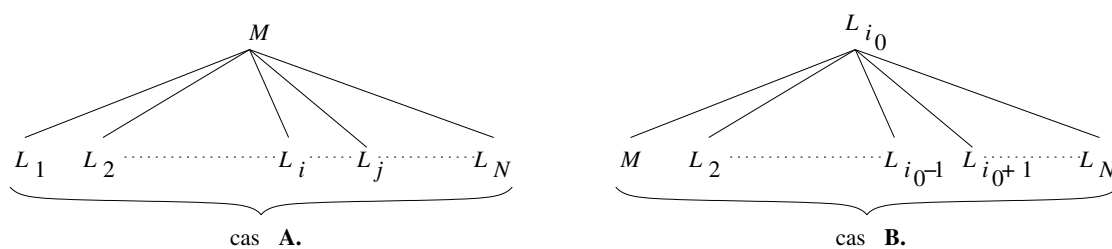


FIG. 7.1 – Situations de dialogue à plusieurs locuteurs

Ces situations sont montrées dans la figure 7.1 ; dans le cas **A.** les structures rhétoriques calculées par la machine, pour ce contexte de dialogue, sont :

$$SDRS_{1NM} = \bigcup_{i=1}^N SDRS_M^{Mi}.$$

Dans le cas **B.** les structures rhétoriques calculées par la machine pour ce contexte de dialogue sont :

$$SDRS_{1NM} = SDRS_M^{Mi_0} \cup \left(\bigcup_{i=1; i \neq i_0} SDRS_M^{i_0i} \right).$$

Dans le chapitre 5 nous avons présenté de manière très détaillée tout ce cadre (que nous reprenons par ailleurs dans le chapitre 6) ; là, nous avons montré comment des dialogues à plusieurs locuteurs, pertinents par rapport aux applications du type service, sont réductibles à des alternations entre les situations de dialogue **A.** et **B.** montrées ci-dessus. Dans le chapitre 5 nous avons aussi vu un mécanisme pour commuter automatiquement entre ces deux situations, tout au long d'un dialogue donné.

Cependant, nous présentons ci-dessous un exemple de dialogue typique relié à la tâche de réservation de livres dans une bibliothèque, pour observer la commutation entre les situations **A.** et **B.** Le dialogue est annoté au niveau des tours de parole (exprimés comme un indice inférieur pour M et un indice supérieur pour les utilisateurs humains), et trois utilisateurs humains L_1 , L_2 et L_3 qui essaient de réserver quelques livres ; la machine est le bibliothécaire virtuel. Pour chaque tour de la machine, la situation de dialogue dans laquelle cette dernière se place est marquée entre parenthèses³ :

M_1 (A) : Bonjour, je suis Groplan, je peux vous aider à chercher des documents dans notre bibliothèque. Vous pouvez, à tout moment, demander des informations sur la disponibilité d'un certain livre, CD ou DVD. Qu'est-ce que je peux faire pour vous ?

L_1^1 : Bonjour, eh ben, nous voudrions une pièce de théâtre, sur un inceste...

L_2^1 : Une tragédie grecque ancienne ou quelque chose comme ça...

M_2 (B) : Un instant... voilà : j'ai trouvé huit tragédies grecques anciennes où un inceste apparaît. Pourriez-vous donner plus de détails, par exemple sur l'auteur ?

L_1^2 : Aristophane ?

L_3^2 : Ou sinon, Virgile ?

M_3 (B) : Je ne peux trouver aucune tragédie ancienne écrite par Aristophane ou Virgile où un inceste apparaît. Néanmoins, j'ai trouvé trois tragédies par Sophocle, une

³Cet exemple est repris, en partie, du chapitre 5.

tragédie par Eschyle et une tragédie par Euripide, où un inceste apparaît. Voulez-vous choisir un de ces auteurs, vous informer sur d'autres auteurs pertinents, ou effectuer une nouvelle recherche ?

L_2^3 : Qu'est-ce que vous avez de Sophocle ?

M_4 (A) : Eh ben, par exemple « Oedipe roi »...

L_1^4 : De quoi s'agit-il dans cette pièce ?

L_3^4 : S'agit-il d'un inceste entre frère et sœur ?

M_5 (B) : En fait, il n'y a pas d'inceste entre frère et sœur dans « Oedipe roi », mais il s'agit, essentiellement, d'un parricide, suivi par un inceste entre mère et fils...

Cet exemple a été analysé de manière détaillée dans d'autres chapitres de cette thèse : dans le chapitre 5 nous avons vu en détail la manière dont la machine se construit une représentation rhétorique de ce dialogue ; dans le chapitre 6 nous avons vu comment les ellipses sémantiques sont calculées pour les tours de parole que la machine produit. Par conséquent, nous n'allons pas reprendre dans ce chapitre ces analyses ; nous avons repris cet exemple ici seulement pour mettre en évidence l'alternance entre les situations **A.** et **B.** de dialogue à plusieurs locuteurs, notamment le fait que ces deux situations « recouvrent » tout ce dialogue, qui représente en même temps un exemple typique d'utilisation de notre cadre théorique.

7.3 Réglage de la force illocutoire en dialogue à deux

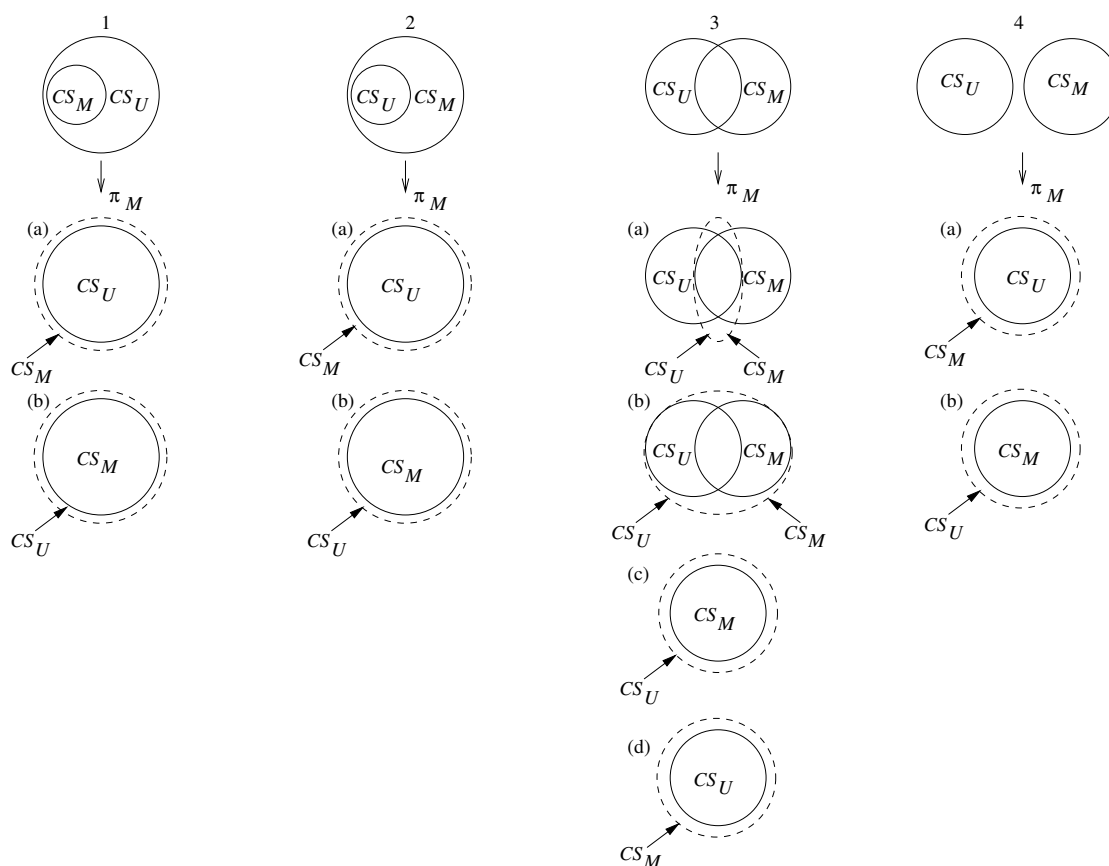
7.3.1 Calcul du degré de force illocutoire

L'idée fondamentale qui guide les mécanismes par lesquels les engagements sont reliés au degré de force illocutoire est que plus les listes d'engagements des interlocuteurs (pour le cas du dialogue classique entre la machine et un locuteur humain, il s'agit de CS_U et CS_M) sont semblables (voire identiques), plus le degré de force illocutoire peut être proche de 0 (zéro). Autrement dit, dans une situation idéale où M et U partagent exactement les mêmes engagements tout au long du dialogue, la machine peut exprimer ces énoncés de manière neutre (au niveau illocutoire, donc en attribuant une force illocutoire zéro à ses contributions dialogiques).

Pour la plupart des tours de parole en dialogue (on désigne par m un tour de parole quelconque), nous avons que $CS_{U|m} \neq CS_{M|m}$, c'est-à-dire, les listes d'engagements de U et M ne sont pas équivalentes⁴ après le tour de parole m . Cette situation revient à plusieurs cas possibles (dans les traitements qui suivent, la notation m du tour de parole sera supprimée) :

1. $CS_M \subset CS_U$, c'est-à-dire, la liste d'engagements de la machine est strictement incluse dans la liste d'engagements de l'utilisateur ; cela est typique pour des dialogues où l'utilisateur apprend de nouvelles tâches à la machine ;
2. $CS_M \supset CS_U$, c'est-à-dire, la liste d'engagements de la machine inclut strictement la liste d'engagements de l'utilisateur ; cela est typique pour les dialogues d'enseignement ;

⁴On considère cette équivalence à un niveau sémantique, revenant à l'identité des modèles où les formules logiques dans les listes d'engagements sont satisfiables.

FIG. 7.2 – Listes d’engagements pour U et M

- $CS_M \cap CS_U \equiv CS \cap \neq \emptyset$, c’est-à-dire, il y a un certain chevauchement entre les listes d’engagements de la machine et de l’utilisateur ; cette situation est typique pour les dialogues de service, lorsque la machine interagit avec des utilisateurs inconnus d’avance ;
- $CS_M \cap CS_U \equiv \emptyset$, c’est-à-dire, les listes d’engagements de la machine et de l’utilisateur n’ont rien en commun ; cela est typique pour les dialogues où les interlocuteurs ont des arrière-plans culturels fondamentalement différents, ou l’utilisateur se moque du système, ou encore l’utilisateur ne joue pas le jeu du dialogue ; ce type de dialogues ne représente que des interactions sans succès.

Avant d’aller plus loin, il faut préciser ce que nous entendons par l’intersection de deux listes d’engagements : cette notion peut être considérée soit à un niveau purement syntaxique (revenant aux formules logiques contenues *ad litteram* dans les deux listes d’engagements), ou à un niveau sémantique (revenant aux formules logiques présentes dans les deux listes d’engagements et satisfiables dans le même ensemble de modèles). Comme nous l’avons fait pour l’équivalence des modèles, nous considérons l’intersection des modèles à un niveau sémantique (une définition syntaxique aurait été trop contraignante). Ainsi, pour pouvoir déterminer automatiquement l’intersection de deux listes d’engagements, ou pour évaluer leur équivalence, nous

devons parcourir un ensemble d'étapes, précisées de manière concise ci-dessous :

pour deux listes d'engagements CS_i et CS_j :

(a) **pour** chaque formule logique $\lambda_k^{(i)} \in CS_i$ et $\lambda_l^{(j)} \in CS_j$:

i. calculer les modèles où ces formules sont satisfiables :

$$\begin{aligned} \mu_k^{(i)}[m] : \mu_k^{(i)}[m] \models \lambda_k^{(i)} ; \\ \mu_l^{(j)}[n] : \mu_l^{(j)}[n] \models \lambda_l^{(j)} ; \end{aligned}$$

ii. calculer les conjonctions de ces modèles :

$$\begin{aligned} M_k^{(i)} &= \bigwedge_{m: \mu_k^{(i)}[m] \neq \emptyset} (\mu_k^{(i)}[m]) ; \\ M_l^{(j)} &= \bigwedge_{n: \mu_l^{(j)}[n] \neq \emptyset} (\mu_l^{(j)}[n]) ; \end{aligned}$$

(b) calculer les réunions des conjonctions des modèles :

$$\begin{aligned} M_i &= \bigcup_{k: M_k^{(i)} \neq \emptyset} (M_k^{(i)}) ; \\ M_j &= \bigcup_{l: M_l^{(j)} \neq \emptyset} (M_l^{(j)}) . \end{aligned}$$

Ensuite, nous comparons ces ensembles M_i et M_j pour les deux listes d'engagements CS_i et CS_j , afin à évaluer leur équivalence ou leur intersection. Le problème crucial avec cette approche est comment déterminer l'ensemble des modèles qui satisfont les formules logiques ; essentiellement, cela se réalise en considérant que l'ontologie de tâches en entier est un modèle, d'où nous éliminons dans un premier temps les prédicats qui rendent la formule logique considérée non-satisfiable dans le modèle. Ensuite, nous pouvons encore générer des modèles où la formule logique est satisfiable, en éliminant encore des prédicats dans l'ontologie de tâches, mais ainsi nous obtenons un ensemble de modèles strictement ordonnés par rapport à l'inclusion. Nous observons donc que les ensembles $M_k^{(i)}$, pour chaque formule logique $\lambda_k^{(i)}$ dans la liste d'engagements CS_i , sont des sous-ensembles de prédicats dans l'ontologie de tâches, tels que l'ajout de $\lambda_k^{(i)}$ à ce sous-ensemble ne mène pas à une contradiction.

Pour ce qui tient à la relation d'inclusion stricte (\subset) entre les listes d'engagements, elle se définit également en termes sémantiques, donc de modèles dans lesquels les formules contenues dans ces listes d'engagements sont satisfiables. Ainsi, une relation d'inclusion entre deux listes d'engagements CS_i et CS_j revient à la relation d'inclusion *inverse* entre les ensembles correspondants de modèles M_i et M_j (voir la discussion ci-dessus) :

$$CS_i \subset CS_j \Leftrightarrow M_j \subset M_i.$$

Cela s'explique par le fait que plus une liste d'engagements contient des formules (modulo l'équivalence logique), moins il y a de modèles qui satisfassent *toutes* ces formules. En ce qui concerne la relation de différence entre deux listes d'engagements, elle est toujours définie en termes de modèles : le résultat d'une différence entre deux listes d'engagements représente l'ensemble des modèles qui ne satisfont aucune formule dans la deuxième liste d'engagements, mais satisfont toutes les formules dans la première liste d'engagements qui ne sont pas équivalentes aux formules dans la deuxième liste d'engagements :

$$CS_i \setminus CS_j = \{ \mu : \forall \lambda_j \in CS_j \mu \not\models \lambda_j \wedge \forall \lambda_i \in CS_i : \lambda_i \not\models \lambda_j \mu \models \lambda_i \}.$$

Du fait que tout au long de cette présentation nous adoptons la perspective de la machine, afin de prédire son comportement dans la production d'un tour de parole, nous supposons que, dans chacun des quatre cas présentés ci-dessus, la contribution dialogique suivante attendue appartient

à la machine ; nous notons ainsi par π_M le tour de parole *suivant* attendu de M . Dans ce contexte, nous analysons de manière détaillée ci-dessous chacune des quatre situations indiquées ci-dessus (concernant les relations entre les listes d'engagements) et illustrées dans la figure 7.2 :

1. Le cas $CS_M \subset CS_U$; le but de M en dialogue est de rendre équivalentes les listes d'engagements des interlocuteurs, c'est-à-dire, $CS_M \equiv CS_U$. Nous supposons que, lorsque la machine produit la contribution π_M , elle essaye de modifier aussi peu que possible sa propre liste d'engagements. De plus, pour la simplicité des traitements, nous supposons que π_M est un seul énoncé ; en fait, cela ne réduit pas la généralité du problème, car nous pouvons supposer que la machine produit séquentiellement chaque énoncé, et ce même dans un tour de parole qui contient plusieurs énoncés. Ainsi, il y a deux situations possibles :

- (a) $CS_M \leftarrow_{\pi_M} CS_M \cup (CS_U \setminus CS_M)$:

Dans ce cas, M doit produire un énoncé π_M *via* lequel elle accepte tous les énoncés antérieurs dans la liste d'engagements de U , mais absents de sa liste d'engagements, c'est-à-dire, M s'engage à $K(\pi_{U-}) = CS_U \setminus CS_M$, où π_{U-} désigne un (ensemble d')énoncé(s) précédemment produit(s) par U . Ainsi, deux calculs sont indépendamment effectués :

- le composant de structuration rhétorique (voir chapitres 3 et 4) choisit une relation rhétorique dialogique de confirmation : $\rho_d^{(c)}(\pi_{U-}, \pi_M)$, et
- une force illocutoire *faible* est choisie pour π_M ⁵ : $\partial\phi(F(K(\pi_M))) = \partial\phi(\pi_M) = -1$.

- (b) $CS_U \leftarrow_{\pi_M} CS_U \setminus (CS_U \setminus CS_M)$:

Dans ce cas, M doit produire l'énoncé π_M *via* lequel elle essaye de convaincre U de retirer son engagement à $K(\pi_{U-})$ (c'est-à-dire, aux faits auxquels la machine ne s'est pas, elle aussi, engagée). Ainsi, deux calculs sont indépendamment effectués :

- le composant de structuration rhétorique choisit une relation rhétorique dialogique de contradiction : $\rho_d^{(-c)}(\pi_{U-}, \pi_M)$, et
- une force illocutoire *très forte* est choisie : $\partial\phi(\pi_M) = 2$.

La machine va d'abord essayer la deuxième possibilité et, seulement si cela échoue (c'est-à-dire, si l'utilisateur ne coopère pas avec la machine pour ajuster sa liste d'engagements à celle du système), la machine choisira la première possibilité.

2. Le cas $CS_M \supset CS_U$; en supposant que la composante de génération « pragmatique » dans la machine a le même but en dialogue qu'au cas précédent (c'est-à-dire, de rendre les listes d'engagements équivalents) il y a aussi deux possibilités :

- (a) $CS_M \leftarrow_{\pi_M} CS_M \setminus (CS_M \setminus CS_U)$:

Dans ce cas, M doit produire l'énoncé π_M *via* lequel elle concède à retirer les énoncés antérieurs auxquels elle s'était engagée, et auxquels l'utilisateur ne s'était pas, lui aussi, engagé (c'est-à-dire, $K(\pi_{M-}) = CS_M \setminus CS_U$, où π_{M-} désigne un (des) énoncé(s) précédemment produit(s) par M). Ainsi, deux calculs sont indépendamment effectués :

⁵Nous allons désormais simplifier la notation et écrire $\partial\phi(\pi)$ au lieu de $\partial\phi(F(K(\pi)), K(\pi))$ pour le degré de force illocutoire d'un énoncé libellé π .

- le composant de structuration rhétorique choisit une relation rhétorique dialogique de confirmation entre π_M et un (des) énoncé antérieur(s) provenu(s) de U (et désigné(s) par π_{U-}), tel(s) que $K(\pi_{U-}) \Rightarrow \neg K(\pi_{M-})$ et qu'il existe une relation rhétorique dialogique de contradiction $\rho_d^{(-c)}(\pi_{M-}, \pi_{U-}) : \rho_d^{(c)}(\pi_{U-}, \pi_M)$, et
- une force illocutoire *très faible* est choisie : $\partial\phi(\pi_M) = -2$.

(b) $CS_U \leftarrow_{\pi_M} CS_U \cup (CS_M \setminus CS_U)$:

Dans ce cas, M doit produire un énoncé π_M via lequel il essaye de convaincre U de s'engager à $K(\pi_{M-}) = CS_M \setminus CS_U$. Ainsi, deux calculs sont indépendamment effectués :

- le composant de structuration rhétorique choisit une relation rhétorique monologique, entre π_M et π_{M-} : $\rho_m(\pi_{M-}, \pi_M)$, et
- une force illocutoire *forte* est choisie pour π_M : $\partial\phi(\pi_M) = 1$.

Comme pour le cas précisé auparavant, la machine essayera d'abord la deuxième possibilité et, seulement si celle-ci échoue, M choisira la première possibilité.

3. Le cas $CS_M \cap CS_U = CS_\cap$. Pour la présentation qui s'ensuit, nous introduisons les notations suivantes : (i) $K(\pi_{M-}) = CS_M \setminus CS_U$, (ii) $K(\pi_{U-}) = CS_U \setminus CS_M$, (iii) $K(\pi_{UM-}) = CS_U \cap CS_M$, (iv) $K(\pi_{M-U-}) = CS_M \setminus CS_\cap$, (v) $K(\pi_{U-M-}) = CS_U \setminus CS_\cap$. Nous avons ainsi que $CS_U \cup CS_M = \{K(\pi_{M-}), K(\pi_{U-}), K(\pi_{UM-})\}$.

Tout en gardant les mêmes suppositions concernant le but de M et l'heuristique concernant les préférences pour la mise à jour des listes d'engagements, il y a quatre possibilités pour ce cas :

(a) $(CS_M \leftarrow_{\pi_M} CS_\cap) \wedge (CS_U \leftarrow_{\pi_M} CS_\cap)$:

Dans cette situation, M doit produire un énoncé π_M via lequel elle concède à ne plus s'engager à $K(\pi_{M-U-})$ et elle essaye de convaincre U de ne plus s'engager à $K(\pi_{U-M-})$. Ainsi, deux calculs sont indépendamment effectués :

- π_M est éclaté en deux énoncés, π'_M et π''_M tels que $K(\pi'_M) \wedge K(\pi''_M) = K(\pi_M)$; alors,
 - via π'_M la machine concède à ne plus s'engager à $K(\pi_{M-U-})$, donc le composant de structuration rhétorique calcule une relation rhétorique monologique $\rho_m(\pi_{M-U-}, \pi'_M)$, telle que $K(\pi'_M) \Rightarrow \neg K(\pi_{M-U-})$;
 - via π''_M la machine essaye de convaincre U de ne plus s'engager à $K(\pi_{U-M-})$, ainsi le composant de structuration rhétorique calcule une relation rhétorique dialogique de contradiction : $\rho_d^{(-c)}(\pi_{U-M-}, \pi''_M)$;
- attribution des degrés de force illocutoire :
 - à π'_M nous attribuons un degré de force illocutoire *très faible* : $\partial\phi(\pi'_M) = -2$;
 - à π''_M nous attribuons un degré de force illocutoire *très fort* : $\partial\phi(\pi''_M) = 2$.

(b) $(CS_M \leftarrow_{\pi_M} CS_M \cup CS_U) \wedge (CS_U \leftarrow_{\pi_M} CS_M \cup CS_U)$:

Dans ce cas, M doit produire l'énoncé π_M via lequel elle concède à s'engager à $K(\pi_{U-})$ et elle essaye de convaincre U de s'engager à $K(\pi_{M-})$. Ainsi, deux calculs sont indépendamment effectués :

- π_M est éclaté en deux énoncés, π'_M et π''_M tels que $K(\pi'_M) \wedge K(\pi''_M) = K(\pi_M)$; alors,
 - via π'_M la machine concède à s'engager à $K(\pi_{U-})$, ainsi le composant de structuration rhétorique calcule une relation rhétorique dialogique de confirmation : $\rho_d^{(c)}(\pi_{U-}, \pi'_M)$;

- via π''_M la machine tente de convaincre U de s'engager à $K(\pi_{M-})$, ainsi le composant de structuration rhétorique calcule une relation rhétorique monologique : $\rho_m(\pi_{M-}, \pi''_M)$;
 - attribution des degrés de force illocutoire :
 - à π'_M nous attribuons un degré de force illocutoire *faible* : $\partial\phi(\pi'_M) = -1$;
 - à π''_M nous attribuons un degré de force illocutoire *fort* : $\partial\phi(\pi''_M) = 1$.
- (c) $CS_U \leftarrow_{\pi_M} CS_M \equiv CS_U \setminus (CS_U \setminus CS_M) \cup (CS_M \setminus CS_U)$:
- Dans ce cas, M doit produire un énoncé π_M via lequel elle essaye de convaincre U de retirer son engagement à $K(\pi_{U-})$ et à s'engager à $K(\pi_{M-})$. Ainsi, deux calculs sont indépendamment effectués :
- π_M est éclaté en deux énoncés, π'_M et π''_M tels que $K(\pi'_M) \wedge K(\pi''_M) = K(\pi_M)$; ensuite, le composant de structuration rhétorique calcule une relation rhétorique dialogique de contradiction entre π'_M et π_{U-} : $\rho_d^{(-c)}(\pi_{U-}, \pi'_M)$ et une relation rhétorique monologique entre π''_M et π_{M-} : $\rho_m(\pi_{M-}, \pi''_M)$;
 - un degré de force illocutoire *très fort* est choisi pour le premier énoncé, et un degré de force *fort*, pour le deuxième énoncé : $\partial\phi(\pi'_M) = 2$ et, respectivement $\partial\phi(\pi''_M) = 1$.
- (d) $CS_M \leftarrow_{\pi_M} CS_U \equiv CS_M \setminus (CS_M \setminus CS_U) \cup (CS_U \setminus CS_M)$:
- Dans ce cas M doit produire un énoncé π_M via lequel elle concède à ne plus s'engager à $K(\pi_{M-U-})$ et à s'engager à $K(\pi_{U-M-})$. Ainsi, les deux calculs suivants sont indépendamment effectués :
- π_M est éclaté en deux énoncés, π'_M et π''_M tels que $K(\pi'_M) \wedge K(\pi''_M) = K(\pi_M)$; alors,
 - via π'_M la machine concède à retirer son engagement à $K(\pi_{M-U-})$, ainsi le composant de structuration rhétorique calcule une relation rhétorique monologique $\rho_m(\pi_{M-U-}, \pi'_M)$ telle que $K(\pi'_M) \Rightarrow \neg K(\pi_{M-U-})$;
 - via π''_M la machine s'engage à $K(\pi_{U-M-})$, ainsi le composant de structuration rhétorique calcule une relation rhétorique dialogique de confirmation : $\rho_d^{(c)}(\pi_{U-M-}, \pi''_M)$;
 - attribution des degrés de force illocutoire :
 - à π'_M nous attribuons un degré de force illocutoire *très faible* : $\partial\phi(\pi'_M) = -2$;
 - à π''_M nous attribuons un degré de force illocutoire *faible* : $\partial\phi(\pi''_M) = -1$.

Etant donnée l'heuristique selon laquelle M essaye de modifier aussi peu que possible sa liste d'engagements, en préférant y ajouter des entités (c'est-à-dire, des formules logiques) plutôt qu'en enlever, l'ordre des préférences pour les possibilités énumérées ci-dessus est la suivante : (c) \rightarrow (b) \rightarrow (a) \rightarrow (d).

4. Le cas $CS_M \cap CS_U = \emptyset$; en adoptant la contrainte selon laquelle on ne permet pas qu'une liste d'engagements soit vide suite à une des opérations de mise à jour, et tout en gardant la même supposition sur le but dialogique de M ainsi que la même heuristique concernant les préférences dans la mise à jour des listes d'engagements, nous avons deux possibilités :

- (a) $CS_M \leftarrow_{\pi_M} CS_U$:

Dans cette situation, via π_M M renonce à tous ses engagements, en acceptant en revanche tous les engagements de U ; ainsi, cela représente une version extrême du cas 3.(d), mais le traitement formel est identique à ce cas-là.

(b) $CS_U \leftarrow_{\pi_M} CS_M$:

Dans ce cas, la machine produit l'énoncé π_M *via* lequel elle essaye de convaincre U de renoncer à tous ses engagements antérieurs, tout en s'engageant à tous les engagements auxquels la machine s'est engagée ; ainsi, cette situation est une version extrême du cas 3.(c), mais le traitement formel est identique à ce cas-là.

Etant donnée l'heuristique selon laquelle M essaye de modifier au minimum sa liste d'engagements, en préférant y ajouter des formules plutôt qu'en enlever, l'ordre des préférences pour les possibilités présentées ci-dessus pour ce cas est : (b) \rightarrow (a).

En ce qui concerne l'attribution des degrés de force illocutoire, plusieurs commentaires s'imposent, afin d'argumenter le choix d'une échelle à cinq niveaux, et de clarifier la distinction entre les degrés de force très fort (valeur +2), fort (valeur +1), neutre (valeur 0), faible (valeur -1) et très faible (valeur -2). Ainsi, étant donné un locuteur L et un énoncé libellé π , produit par L , nous pouvons mettre en évidence plusieurs situations :

- lorsque *via* π , L concède à enlever quelque chose de sa liste d'engagements CS_L , comme $CS_L \leftarrow_{\pi} CS_L \setminus \{\pi_{L-}\}$ pour quelques énoncés antérieurs π_{L-} de L , nous avons que $\partial\phi(\pi) = -2$, donc un degré de force illocutoire **très faible** ;
- lorsque *via* π , L ajoute quelque chose à sa liste d'engagements CS_L , comme $CS_L \leftarrow_{\pi} CS_L \cup \{\pi_{L-}\}$ pour quelques énoncés antérieurs d'un autre locuteur, \bar{L} , nous avons que $\partial\phi(\pi) = -1$, donc un degré de force illocutoire **faible** ;
- lorsque *via* π , L essaye de convaincre un autre locuteur \bar{L} d'ajouter quelque chose à sa liste d'engagements $CS_{\bar{L}}$ (c'est-à-dire, $CS_{\bar{L}} \leftarrow_{\pi} CS_{\bar{L}} \cup \{\pi_{L-}\}$), nous avons que $\partial\phi(\pi) = 1$, donc un degré de force illocutoire **fort** ;
- lorsque *via* π , L essaye de convaincre un autre locuteur \bar{L} d'enlever quelque chose de sa liste d'engagements $CS_{\bar{L}}$ (c'est-à-dire, $CS_{\bar{L}} \leftarrow_{\pi} CS_{\bar{L}} \setminus \{\pi_{L-}\}$ pour un énoncé antérieur π_{L-} de \bar{L}), nous avons que $\partial\phi(\pi) = 2$, donc un degré de force illocutoire **très fort**.

Pour le degré de force illocutoire **neutre** (c'est-à-dire, de valeur 0), nous stipulons qu'un énoncé peut être produit (par la machine) avec un degré de force neutre seulement lorsqu'il asserte des faits incontestables sur le monde (faits qui sont codés dans l'ontologie de tâches comme des lois naturelles par exemple – voir chapitre 8 pour une définition de ce type de connaissances), par exemple le fait que 'Un livre ne peut normalement pas être lu en l'absence totale de la lumière', il réalise l'acte de langage du type F^S *et* est prononcé pour la première fois dans le dialogue (c'est-à-dire, il n'est pas réaffirmé dans un essai de faire l'allocutaire changer d'avis et accepter l'énoncé, après l'avoir rejeté auparavant). Par conséquent, tout autre type d'énoncé, réalisant tout autre type d'acte de langage, ne peut pas avoir un degré de force illocutoire neutre.

Ainsi, nous observons comment le jeu des relations possibles entre les listes d'engagements de la paire d'interlocuteurs en dialogue est mis en correspondance avec un ensemble de valeurs pour le degré de force illocutoire, ce qui offre une sorte de justification *a posteriori* pour l'échelle à cinq niveaux pour ce degré de force. Même si le choix d'une échelle discrète pour les degrés de force illocutoire peut paraître plutôt un choix méthodologique, étant donnée la possibilité d'annoter plus facilement des corpus de structures linguistiques figées (éventuellement à trous) pour établir la correspondance entre ces degrés de force illocutoire et les diverses réalisations linguistiques d'une intention communicationnelle (spécifiée, nous le rappelons, sous forme logique), nous remarquons que cette échelle quintuple reflète bien les diverses relations (en termes ensem-

blistes) entre les listes d'engagements des interlocuteurs. Par exemple, si nous avons choisi une échelle continue de valeurs dans un intervalle (par exemple, $[-1; +1]$), nous aurions eu du mal à mettre en correspondance ces valeurs avec les relations entre les listes d'engagements, sauf si nous avons assigné des « degrés de réalisation » aux relations entre les listes d'engagements (nous aurions pu dire que deux listes d'engagements se recouvrent plus ou moins, au sens où la cardinalité de l'ensemble des modèles qui satisfont les formules dans les deux listes d'engagements est plus ou moins importante). De toute manière, une telle modélisation ne nous semble pas être appuyée par les exemples de dialogues que nous avons considérés, tout au long de cette thèse.

Une autre critique potentielle de l'approche proposée ici concerne le critère choisi pour le succès des dialogues. En effet, nous pourrions considérer l'exemple de dialogue suivant, entre la machine, M (un bibliothécaire virtuel) et un utilisateur, L :

U_1 : Bonjour, je voudrais des livres sur la Révolution française !

M_1 : Bonjour, Monsieur, je peux vous proposer deux livres : un livre introductif, 'X' et un autre, plus avancé, 'Y'.

U_2 : Je voudrais réserver le livre 'Y', cela me paraît intéressant.

M_2 : OK, donnez-moi votre carte s'il vous plaît.

Dans ce dialogue, la proposition qu'« il existe dans la bibliothèque un livre 'Y' sur la Révolution française » devrait faire partie de CS_M et de CS_U . Néanmoins, tandis que la proposition « il existe dans la bibliothèque un livre 'X' sur la Révolution française » fait partie de CS_M , U ne s'est pas engagé à son contenu (de manière implicite ou explicite) ; U est neutre à l'égard du contenu de cette proposition ; donc la proposition ne fait pas partie de CS_U . Ainsi, après ces tours de parole, nous avons que $CS_U \subset CS_M$, pourtant, le dialogue a du succès, puisque l'utilisateur arrive à faire sa réservation. Le dialogue n'aurait pas eu plus de succès si, par exemple, U avait produit aussi « OK, d'accord » au début du tour de parole U_2 , pour s'engager au contenu de tout le tour de parole antérieur de M (c'est-à-dire, à tout le contenu de M_1) :

U'_2 : OK, d'accord. Je voudrais réserver le livre 'Y', cela me paraît intéressant.

D'un côté, l'équivalence des listes d'engagements des interlocuteurs en dialogue représente une condition *suffisante* au succès du dialogue ; même si cette condition n'est donc pas toujours *nécessaire*, son accomplissement assure que le dialogue a du succès. De l'autre côté, le tour de parole U_2 peut constituer une confirmation *implicite indirecte* (essentiellement, il s'agit du fait que si U ne contredit, de manière explicite ou implicite, aucune partie du tour précédent de M , alors nous posons que *par défaut* U s'engage à tout ce tour de parole de M) du tour *entier* M_1 de la machine.

Enfin, une autre critique qui nous paraît pertinente est que le degré de force illocutoire n'est pas déterminé que par la structure rhétorique et les engagements des interlocuteurs ; il y a aussi les rôles sociaux, et leur positions relatives dans la société qui jouent un rôle important pour limiter l'ensemble des valeurs que le degré de force illocutoire peut prendre. Selon notre cadre, comme présenté jusqu'à maintenant, dans une situation où l'utilisateur U affirme α et la machine M affirme $\neg\alpha$, si la machine souhaite convaincre U de renoncer à s'engager à α , alors elle devrait affirmer $\neg\alpha$ avec un degré de force ($\partial\phi$) de +2. Cela amènerait la machine à produire un énoncé :

‘Mais Monsieur, α n’est certainement pas vraie, vous comprenez ?’. Dans un scénario de service, où U est le client et M le « serveur », cet énoncé ne serait pas approprié, puisqu’il amènerait U à se plaindre du manque de politesse de M . Cela est un cas où le pouvoir de M est beaucoup plus réduit que celui de U dans l’interaction. Tout simplement, M n’a pas l’autorité de produire des énoncés avec une force illocutoire de +2. Tout ce que la machine peut faire dans ce cas est produire un énoncé avec un degré de force $\partial\phi = +1$: ‘Monsieur, je suis convaincu que α n’est pas vraie’. Nous observons donc comment un statut social « inférieur » par rapport à l’allocutaire induit une marge supérieure plus serrée pour le degré de force illocutoire. Etant donné cela, nous pouvons nous poser la question si, inversement, il existe des configurations relationnelles qui induisent aussi une marge inférieure plus serrée au degré de force illocutoire. Nous croyons qu’ici une telle marge inférieure ne peut être imposée que par la personnalité du locuteur : par exemple, si celui-ci a une nature fière, alors il aura tendance à ne pas produire des énoncés ayant un degré de force de -2 (pour ainsi renoncer explicitement à ses engagements antérieurs), en préférant, au lieu de cela, produire des énoncés d’un degré de force de -1 , *via* lesquels il concède à accepter les énoncés de l’allocutaire (qui pourraient à leur tour *impliquer* l’abandon de certains de ses engagements) : par exemple, considérons le dialogue suivant entre un utilisateur et la machine, qui est une machine « fière », au sens précisé ci-dessus :

U_1 : Mais le livre ‘X’ ne se trouve pas au premier étage, comme vous m’aviez dit !
Il se trouve en fait au deuxième étage !

M_1 : Hm... je crois que vous avez raison ; j’enverrai donc un de mes collègues au deuxième étage pour vous apporter le livre.

En M_1 (qui a une force illocutoire de -1), la machine n’abandonne pas de manière explicite son engagement antérieur (présumé) au fait que le livre ‘X’ se trouvait au premier étage, en préférant confirmer de manière floue le tour U_1 de l’utilisateur. Si la machine avait été moins « fière », elle aurait explicitement renoncé à son engagement au fait que le livre ‘X’ se trouvait au premier étage, comme dans un tour de parole $M_1^{(o)}$: ‘Oui, effectivement, je me suis trompé, désolé ; j’enverrai donc un collègue au deuxième étage pour vous apporter le livre’.

Pour conclure sur cet aspect des relations sociales, nous proposons, dans une première approximation, que le domaine des valeurs du degré de force illocutoire soit limité supérieurement par un statut social « inférieur » à l’interlocuteur : dans l’exemple du bibliothécaire virtuel, celui-ci ne pourra produire que des énoncés ayant un degré de force dans l’ensemble $\{-2; -1; 0; +1\}$; l’utilisateur humain pourra en revanche produire des énoncés avec des degrés de force illocutoire de -2 à $+2$. Bien entendu, la question de préciser de manière plus fine la relation entre les statuts sociaux des interlocuteurs et le domaine des valeurs pour les degrés de force illocutoire avec lesquels ils peuvent produire des énoncés en interaction, ainsi que la manière dont la *relation sociale* entre ces interlocuteurs contraint les choix des degrés de force illocutoire reste ouverte à des études futures.

7.3.2 Exemples en dialogue homme-machine

Dans ce paragraphe nous allons considérer deux exemples de dialogues, pour illustrer le mécanisme d’ajustement du degré de force illocutoire en dialogue entre la machine et un seul interlocuteur ; la correspondance entre le degré de force illocutoire calculé et la forme de surface

concrète ne fait pas l'objet de ce chapitre et, par ailleurs, constitue une piste pour des recherches futures. Cependant, nous allons présenter dans l'annexe A un travail préliminaire dans ce sens, pour mettre en correspondance le degré de force illocutoire avec la forme de surface, pour un contenu propositionnel donné (forme logique) de l'énoncé.

Ces deux dialogues ne se distinguent que par certains tours de parole, le contexte de la tâche étant le même : un utilisateur U essaye de trouver un livre 'B' dans une bibliothèque, dont le bibliothécaire virtuel est la machine M :

Dialogue 1 :

U_1 : Excusez-moi, pourriez-vous me dire, s'il vous plaît, où je peux trouver le livre 'B' ? –

$\text{F}^{\text{FS}}(\pi_1)$

M_1 : Bonjour, eh ben, vous pouvez trouver le livre 'B' juste au fond du couloir, à gauche. –

$\text{F}^{\text{S}}(\pi_2) ; \text{QAP}(\pi_1, \pi_2) ; \partial\phi(\pi_2) = 1$

U_2 : Mais je viens de passer par là, et je n'ai pas trouvé le livre ! –

$\text{F}^{\text{S}}(\pi_3) ; K(\pi_3) \Rightarrow \neg K(\pi_2) ; P - \text{Corr}(\pi_2, \pi_3)$

M_2 : Ô, je me suis trompé, désolé ; effectivement, vous pouvez trouver le livre 'B' au premier étage, à gauche. –

$\text{F}^{\text{S}}(\pi_4) ; \text{Contrast}(\pi_2, \pi_4) ; P - \text{Elab}(\pi_3, \pi_4) ; \text{QAP}(\pi_1, \pi_4) ; \partial\phi(\pi_4) = -2$

Dialogue 2 :

U_1 : Excusez-moi, pourriez-vous me dire, s'il vous plaît, où je peux trouver le livre 'B' ? –

$\text{F}^{\text{FS}}(\pi_1)$

M'_1 : Bonjour, attendez un instant... voilà : vous pouvez trouver le livre 'B' au premier étage, à gauche. –

$\text{F}^{\text{S}}(\pi'_2) ; \text{QAP}(\pi_1, \pi'_2) ; \partial\phi(\pi'_2) = -1$

U_2 : Mais je viens de passer par là, et je n'ai pas trouvé le livre ! –

$\text{F}^{\text{S}}(\pi_3) ; K(\pi_3) \Rightarrow \neg K(\pi'_2) ; P - \text{Corr}(\pi'_2, \pi_3)$

M'_2 : Ecoutez, vous pouvez certainement trouver le livre 'B' au premier étage, juste à gauche, je viens de le vérifier dans ma base de données ! –

$\text{F}^{\text{S}}(\pi'_4) ; P - \text{Corr}(\pi_3, \pi'_4) ; \text{QAP}(\pi_1, \pi'_4) ; \partial\phi(\pi_4) = 2$

Sous chaque tour de parole dans ces dialogues nous avons marqué le type de l'acte de langage, l'étiquette de l'énoncé et les relations rhétoriques qui relient ce tour de parole aux tours de parole antérieurs⁶ ; de plus, pour les tours de la machine nous avons aussi marqué le degré de force illocutoire. Les relations rhétoriques sont calculées selon les principes décrits dans les chapitres 3 et 4, tandis que l'interpréteur pragmatique dans l'architecture de dialogue (cf. chapitre 1) calcule le type d'acte de langage pour chaque énoncé de l'utilisateur, et le contrôleur de dialogue précise ce type d'acte pour chaque énoncé de la machine. C'est pourquoi nous ne

⁶Pour alléger la présentation, nous considérons que chaque tour de parole ne contient qu'un énoncé.

décrivons plus loin que le processus de mise à jour des listes d'engagements et la manière dont ces listes déclenchent et pilotent le calcul des degrés de force illocutoire. Ces processus sont détaillés ci-dessous, pour chacun des dialogues :

Dialogue 1 :

1. lorsque U produit π_1 , sa liste d'engagements reste non-modifiée, aussi que la liste d'engagements de M : $CS_U \leftarrow_{\pi_1} CS_U \wedge CS_M \leftarrow_{\pi_1} CS_M$;
2. lorsque M produit π_2 , la liste d'engagements de la machine est mise à jour, tandis que la liste d'engagements de U reste, pour l'instant, non-modifiée : $CS_M \leftarrow_{\pi_2} CS_M \cup \{K(\pi_2), \Sigma_{QAP(\pi_1, \pi_2)}\} \wedge CS_U \leftarrow_{\pi_2} CS_U$; le but de M est que U mette à jour sa liste d'engagements avec $K(\pi_2)$ aussi : $CS_U \leftarrow_{\pi_2} CS_U \cup \{K(\pi_2)\}$, donc M attribue une valeur de +1 au degré de force illocutoire de π_2 : $\partial\phi(\pi_2) = +1$;
3. lorsque U produit π_3 , sa liste d'engagements est mise à jour en ajoutant $\neg K(\pi_2)$, tandis que les engagements de M ne changent pas pour l'instant : $CS_U \leftarrow_{\pi_3} CS_U \cup \{\neg K(\pi_2), \Sigma_{P-Corr(\pi_2, \pi_3)}\} \wedge CS_M \leftarrow_{\pi_3} CS_M$;
4. lorsque M produit π_4 , sa liste d'engagements est mise à jour en enlevant $K(\pi_2)$ et en y ajoutant $K(\pi_4)$, tandis que les engagements de U ne changent pas pour l'instant : $CS_M \leftarrow_{\pi_4} CS_M \setminus \{K(\pi_2)\} \cup \{K(\pi_4), \Sigma_{Contrast(\pi_2, \pi_4)}, \Sigma_{P-Elab(\pi_3, \pi_4)}, \Sigma_{QAP(\pi_1, \pi_4)}\}$; le but de M est que la liste d'engagements de U soit mise à jour en y ajoutant $K(\pi_4)$: $CS_U \leftarrow_{\pi_4} CS_U \cup \{K(\pi_4)\}$ ainsi que $K(\pi_4) \Rightarrow \neg K(\pi_2)$, donc M attribue une valeur de -2 au degré de force illocutoire de π_4 : $\partial\phi(\pi_4) = -2$.

Dialogue 2 :

1. lorsque U produit π_1 , sa liste d'engagements reste non-modifiée, aussi que la liste d'engagements de M : $CS_U \leftarrow_{\pi_1} CS_U \wedge CS_M \leftarrow_{\pi_1} CS_M$;
2. lorsque M produit π'_2 , la liste d'engagements de la machine est mise à jour, tandis que la liste d'engagements de U ne change pas pour l'instant : $CS_M \leftarrow_{\pi'_2} CS_M \cup \{K(\pi'_2), \Sigma_{QAP(\pi_1, \pi'_2)}\} \wedge CS_U \leftarrow_{\pi'_2} CS_U$; le but de M est que la liste d'engagements de U soit mise à jour avec $K(\pi'_2)$ aussi : $CS_U \leftarrow_{\pi'_2} CS_U \cup \{K(\pi'_2)\}$; mais, tout d'abord, M doit mettre à jour sa propre liste d'engagements avec $K(\pi'_2)$, donc M attribue une valeur de -1 au degré de force illocutoire de π_2 : $\partial\phi(\pi_2) = -1$ (à la différence du dialogue précédent, M n'a pas $K(\pi'_2)$ dans sa liste d'engagements, donc elle doit demander cette information au contrôleur de tâche) ;
3. lorsque U produit π_3 , sa liste d'engagements est mise à jour en y ajoutant $\neg K(\pi'_2)$, tandis que les engagements de M ne changent pas pour l'instant : $CS_U \leftarrow_{\pi_3} CS_U \cup \{\neg K(\pi'_2), \Sigma_{P-Corr(\pi'_2, \pi_3)}\} \wedge CS_M \leftarrow_{\pi_3} CS_M$;
4. lorsque M produit π'_4 , sa liste d'engagements continue à contenir $K(\pi'_2)$, tandis que les engagements de U ne changent pas pour l'instant : $CS_M \leftarrow_{\pi'_4} CS_M \cup$

$\{\Sigma_{P-Corr}(\pi_3, \pi'_4), \Sigma_{QAP}(\pi_1, \pi'_4)\}$; le but de M est que la liste d'engagements de U soit mise à jour en enlevant $\neg K(\pi'_2)$ et en y ajoutant $K(\pi'_2)$: $CS_U \leftarrow_{\pi'_4} CS_U \setminus \{\neg K(\pi'_2)\} \cup \{K(\pi'_2)\}$, donc M attribue une valeur de +2 au degré de force illocutoire de π'_4 : $\partial\phi(\pi'_4) = 2$.

Pour ces deux dialogues, nous avons été dans deux cas différents, concernant la relation (ensembliste) entre les listes d'engagements de la machine et de l'utilisateur :

- Pour le dialogue 1 nous sommes dans le cas $CS_U \cap CS_M = CS_{\cap} \neq \emptyset$ après que le tour de parole M_2 ait été produit ;
- Pour le dialogue 2 nous sommes dans le cas $CS_U \cap CS_M = \emptyset$ après que le tour de parole M'_2 ait été produit.

Si'il n'y avait pas eu de réglage du degré de force illocutoire, nous aurions toujours obtenu un seul tour de parole, $M^{(0)}$, au lieu de M_2 , ou M'_1 et M'_2 ; le tour de parole $M^{(0)}$ aurait été un énoncé du type :

$M^{(0)}$: Vous pouvez trouver le livre 'B' au premier étage, à gauche.

Cela aurait amené la machine à agencer un dialogue beaucoup moins naturel et ennuyeux.

7.4 Extensions vers le dialogue à plusieurs locuteurs

Le dialogue à plusieurs locuteurs, même lorsqu'il est restreint au cas où la machine participe aux conversations et seul son point de vue est considéré ici, peut avoir lieu dans des contextes d'interaction très généraux. Cependant, nous pensons que pour les dialogues (à plusieurs locuteurs) orientés service, il y a un ensemble limité de cas particulièrement pertinents :

1. la machine M ne parle qu'à un locuteur, L_i , et les autres locuteurs, $\{L_j : j \in \{1, \dots, N\} \setminus \{i\}\}$ n'écoutent pas ; ce cas est réductible au dialogue classique, à deux locuteurs ;
2. M parle à un locuteur, L_i , et les autres écoutent ; dans ce cas il y a plusieurs situations :
 - (a) les autres locuteurs ne sont que des témoins à la discussion entre M et L_i ; cette situation est, elle aussi, réductible au dialogue classique à deux locuteurs ;
 - (b) les autres locuteurs doivent être informés sur ce dont M et L_i discutent ; dans ce cas, il y a plusieurs choix :
 - i. M pourrait générer un acte de langage pour le locuteur le moins informé (c'est-à-dire, un énoncé tel que même le locuteur le moins informé – par exemple, celui dont la liste d'engagements a la cardinalité la moins importante – peut comprendre) ; dans ce cas, le degré de force illocutoire pour cet acte devrait être ajusté comme si le dialogue n'avait lieu qu'entre la machine et ce locuteur le moins informé ;
 - ii. M pourrait générer un acte pour tout le groupe de locuteurs ; du point de vue de la force illocutoire, cela revient à choisir, pour cet acte, un degré de force qui est une fonction des degrés de force calculés pour chacun des membres du groupe, y compris L_i ; par exemple, cette fonction pourrait être le maximum de tous les degrés de force illocutoire, pour les participants au dialogue (voir plus loin) ;

- iii. M pourrait générer un acte destiné seulement à L_i , avec des ajouts spécifiques pour les autres membres du groupe de N locuteurs ; du point de vue de la force illocutoire, cela revient à deux options :
- agir comme au cas 2.(b)(ii.) ci-dessus, avec la seule différence qu'un poids plus important devrait être donné au degré de force calculé selon la liste d'engagements de L_i ;
 - éclater l'énoncé produit par M en plusieurs énoncés : un énoncé (principal) destiné à L_i (exprimé avec un degré de force illocutoire calculé conformément à la liste d'engagements de ce locuteur), et d'autres énoncés, reliés au premier et destinés aux autres locuteurs dans le groupe de N ;

3. M parle à plusieurs locuteurs $\{L_j : j \in J \subset \{1, \dots, N\}\}$; ce cas est réductible à la situation 2.(b)(ii.) ci-dessus, à l'exception du groupe, qui est, cette fois-ci, un *sous-ensemble* du groupe entier de N locuteurs.

Une perspective plus complète et systématique sur ces situations diverses a été présentée dans le chapitre 5, où une formalisation est également proposée et illustrée *via* des exemples. Pour les buts de ce chapitre, nous nous limitons aux situations **A.** et **B.**, montrées dans le chapitre 5 ; ces situations correspondent en fait aux cas montrés ci-dessus, ainsi :

- la situation **A.** correspond à l'option 2.(b)(ii.) ci-dessus ;
- la situation **B.** correspond soit à l'option 1. (avec la précision que M prend en compte, le cas échéant, ce que les autres avaient dit auparavant dans le dialogue) soit à l'option 2.(b).(iii) ci-dessus.

Pour les situations de dialogue à plusieurs locuteurs du chapitre 5, le calcul du degré de force illocutoire pour un énoncé censé être généré par la machine est géré de la même manière que pour le dialogue classique à deux locuteurs, à l'exception du processus de mise à jour des listes d'engagements des partenaires de dialogue, et de la situation où un énoncé produit par la machine est destiné à *plusieurs* interlocuteurs.

Ainsi, la mise à jour des listes d'engagements pour les locuteurs humains, après un énoncé produit par la machine, a lieu comme suit :

1. M produit un énoncé libellé π et destiné au locuteur L_{i_0} ; formellement, cela peut s'écrire comme suit :

$$\text{equals}(M, \text{emitter}(\pi)) \wedge \text{dest}(L_{i_0}) \in \tilde{K}(\pi),$$
où $\text{dest}/1$ est un prédicat qui est vrai si son argument est un destinataire d'un certain énoncé (ce prédicat apparaît dans la forme logique de cet énoncé) et $\tilde{K}(\pi) ::= \text{Act}(\text{dest}(L_{i_0}) \wedge K(\pi))$; $\text{Act}/1$ désigne le type de l'acte de langage réalisé *via* l'énoncé π ⁷ ;
2. la liste d'engagements de L_{i_0} , notée par CS_{i_0} , est mise à jour comme en dialogue classique à deux locuteurs, à moins qu'il n'y ait un **retour indirect** (voir ci-dessous) de celui-ci, par rapport à π ;
3. les listes d'engagements des locuteurs L_i , $i \neq i_0$, désignées par CS_i , sont mises à jour comme suit :

⁷Act/1 représente donc un « recouvrement » (en anglais, « wrapper ») de la fonction F .

- (a) s'il existe un énoncé π' produit par L_i , tel qu'il existe une relation rhétorique dialogique ρ_d telle que $\rho_d(\pi, \pi') \in SDRS_M^{Mi} \wedge \text{equals}(L_i, \text{emitter}(\pi'))$, alors :
- si ρ_d est une relation rhétorique de confirmation ($\rho_d \equiv \rho_d^{(c)}$), alors :
 $CS_i \leftarrow_{\pi} CS_i \cup \{K(\pi); K(\pi')\}$ (les crochets dénotent des entités optionnelles) ;
 - sinon (si ρ_d est une relation rhétorique dialogique de contradiction – $\rho_d \equiv \rho_d^{(-c)}$), alors :
 $CS_i \leftarrow_{\pi, \pi'} CS_i \cup \{\neg K(\pi); K(\pi'); \Sigma_{\rho_d(\pi, \pi')}\}$;
- (b) si pour tout énoncé π' tel que $\text{equals}(L_i, \text{emitter}(\pi')) \wedge \text{greater}(t(\pi'), t(\pi))$, il n'y a aucune relation rhétorique $\rho_d \in SDRS_M^{Mi}$ qui relie l'énoncé π' à π , alors :
 $CS_i \leftarrow_{\pi} CS_i \wedge CS_i \leftarrow_{\pi'} CS_i \cup \{K(\pi')\}$;
- (c) s'il n'y a aucun énoncé produit par L_i et conséquent à l'énoncé π , alors la liste CS_i reste non-modifiée ;
4. si L_{i_0} produit un *retour indirect* à l'énoncé π de M (et destiné à ce locuteur humain), alors cela revient aux étapes suivantes :
- (a) un autre locuteur L_i , $i \neq i_0$ produit π' tel qu'il existe une relation rhétorique dialogique $\rho_d \in SDRS_M^{Mi}$ qui relie π' à π et que $\text{greater}(t(\pi'), t(\pi))$; dans ce cas, nous pouvons nous trouver dans une des situations suivantes :
- si ρ_d est une relation rhétorique dialogique de confirmation ($\rho_d^{(c)}$), alors la liste d'engagements de L_i est mise à jour comme suit :
 $CS_i \leftarrow_{\pi'} CS_i \cup \{K(\pi); \Sigma_{\rho_d(\pi, \pi')}[; K(\pi')]\}$;
 - si ρ_d est une relation rhétorique dialogique de contradiction ($\rho_d^{(-c)}$), alors la liste d'engagements de L_i est mise à jour comme suit :
 $CS_i \leftarrow_{\pi'} CS_i \cup \{\neg K(\pi); \Sigma_{\rho_d(\pi, \pi')}[; K(\pi')]\}$;
- (b) le locuteur L_{i_0} produit l'énoncé π'' tel qu'il existe une relation rhétorique $\bar{\rho}_d \in SDRS_M^{i_0i}$ et que $\text{greater}(t(\pi''), t(\pi'))$; cela revient aux alternatives suivantes :
- si $\bar{\rho}_d$ est une relation rhétorique de confirmation ($\bar{\rho}_d^{(c)}$), alors :
 $CS_{i_0} \leftarrow_{\pi''} CS_{i_0} \cup \{K(\pi') \wedge \Sigma_{\bar{\rho}_d(\pi, \pi')}[; K(\pi'')]\}$;
 - si $\bar{\rho}_d$ est une relation rhétorique de contradiction ($\bar{\rho}_d^{(-c)}$), alors :
 $CS_{i_0} \leftarrow_{\pi''} CS_{i_0} \cup \{\neg K(\pi') \wedge \Sigma_{\bar{\rho}_d(\pi', \pi'')}\}$.

En fait, cette notion de réaction indirecte généralise la notion de confirmation indirecte, telle qu'elle a été présentée dans le chapitre 6, donc nous n'ajouterons pas plus de commentaires à cet égard, en préférant considérer deux exemples de dialogues à trois locuteurs (dont un est la machine) qui illustrent les situations où le destinataire d'un énoncé produit par la machine offre à son tour une réaction indirecte à cet énoncé. Les deux dialogues, impliquant la machine M et deux utilisateurs humains L_1 et L_2 , sont différents en ce qui concerne le degré de force illocutoire pour un énoncé produit par la machine, pour les différentes situations de mise à jour des listes d'engagements :

Dialogue 1 :

M_1 : Je peux vous donner ce livre si vous voulez !

L_1^1 : Allez, ce livre n'est plus sur la rangée !

L_2^1 : Oui, c'est vrai !

M_2 : Mais non, il est là ! ($\partial\phi = +1$)

Dans ce dialogue il y a une relation rhétorique de contradiction entre les tours de parole M_1 et L_1^1 , suivie par une relation rhétorique dialogique de confirmation entre les tours de parole L_1^1 et L_2^1 , suivie par une relation rhétorique dialogique de contradiction entre les tours L_2^1 et M_2 . Par conséquent, en vertu de la discussion menée ci-dessus, le degré de force illocutoire choisi pour réaliser le tour de parole M_2 (un F^S) est $\partial\phi = 1$.

Dialogue 2 :

M_1 : Je peux vous donner ce livre si vous voulez !

L_1^1 : Ô, mais j'en ai vu l'autre fois un autre, 'Y', est-ce que vous l'avez toujours ?

L_2^1 : Oui, effectivement, nous aimerions plutôt ce livre 'Y' !

M_2 : Mais il n'y a certainement pas ce livre ici, Messieurs ! ($\partial\phi = +2$)

Dans ce dialogue il y a les mêmes types de relations rhétoriques que dans le dialogue 1, entre les tours de parole correspondants, mais cette fois-ci la mise à jour des listes d'engagements se passe différemment, puisque M essaye de déterminer la rétraction du contenu asserté dans le tour L_1^1 , des listes d'engagements de L_1 et L_2 . Par conséquent, le degré de force illocutoire choisi pour le tour de parole M_2 (toujours un F^S) est $\partial\phi = +2$ cette fois-ci.

Lorsque M destine explicitement un énoncé π à plusieurs interlocuteurs⁸, désignés par $\{L_i : i \in I\}$, pour un ensemble I précisé par le contrôleur de dialogue (cf. chapitre 5), le degré de force illocutoire est calculé comme une fonction de la valeur maximale des valeurs absolues des degrés de force illocutoire, calculées pour chaque interlocuteur, considéré de manière isolée :

$$\partial\phi(\pi) = \max_{i \in I} (|\partial\phi(\pi)|_{L_i}) \cdot \sigma,$$

où $\partial\phi(\pi)|_{L_i}$ est le degré de force illocutoire calculé pour le locuteur L_i considéré de manière isolée, et σ est le *signe* de la force illocutoire, précisé par :

$$\sigma = \begin{cases} -1, & \text{si } CS_M \leftarrow_{\pi} CS_M \setminus \{K(\pi_-)\} / \exists \pi_- : \text{equals}(\text{emitter}(\pi_-), M) \wedge \text{smaller}(t(\pi_-), t(\pi)); \\ +1, & \text{sinon} \end{cases}$$

De manière informelle, le degré de force illocutoire est négatif lorsque *via* π la machine enlève un énoncé antérieur π_- de sa liste d'engagements, et +1 sinon.

7.5 Discussion et conclusions

Dans ce chapitre nous avons montré que le réglage du degré de force illocutoire est un aspect crucial lorsqu'il s'agit de générer les énoncés de la machine, dans des dialogues. Ainsi, nous avons proposé un mécanisme computationnel pour calculer le degré de force illocutoire ; ce mécanisme est basé sur la structure rhétorique du dialogue et sur les engagements publics des interlocuteurs (plus précisément, un ordinateur et plusieurs utilisateurs humains d'un service mis en place à l'aide de cet ordinateur). Ce mécanisme est intégré à un module de génération pragmatique en dialogue homme-machine à plusieurs locuteurs.

⁸Un énoncé est destiné *explicitement* à un ensemble de locuteurs si dans la formule logique qui exprime le contenu de cet énoncé il existe une conjonction de prédicats $\text{dest}/1$ qui spécifie l'ensemble des destinataires de cet énoncé.

Cependant, l'approche décrite dans ce chapitre a plusieurs limites et aspects qui sont sujet de débat : (i) des six composantes d'une force illocutoire (en tant que définies par Vanderveken [185]), nous ne prenons en compte de manière explicite que trois : le contenu propositionnel (utile tout d'abord pour calculer les relations rhétoriques entre les actes de langage), le degré de force (dont le calcul, pour la génération, fait l'objet de ce chapitre), et le point illocutoire (c'est-à-dire, le type de l'acte de langage) ; (ii) les conditions préparatoires de la force illocutoire sont prises en compte par le contrôleur de dialogue, lorsque celui-ci décide quel type d'acte de langage attribuer à une intention communicationnelle, le mode d'accomplissement est quantifié, au niveau du contrôle de dialogue, par plusieurs stratégies de dialogue (voir notamment les travaux de Caelen [35]), tandis que les conditions de sincérité sont embarquées dans la logique épistémique utilisée par le contrôleur de dialogue [35] ; donc, aucun de ces derniers aspects n'est géré au niveau de la génération. Les six composantes de la force illocutoire sont distribuées, pour la génération, entre le contrôleur du dialogue et la couche de « génération pragmatique » qui fait l'objet de cette thèse ; (iii) un autre point discutable, bien qu'en accord avec Vanderveken ([185], p. 120) est le domaine de valeurs pour le degré de force illocutoire (voir aussi la discussion antérieure sur ce point) : $\{-2, -1, 0, 1, 2\}$; Vanderveken ne limite pas le nombre des degrés de force possibles, mais nous le faisons, pour des raisons méthodologiques (reliés à la possibilité d'annoter de manière convenable un ensemble de structures linguistiques figées, utilisables en génération de surface).

Ces limites générales mises à part, le cadre que nous avons proposé dans ce chapitre a aussi d'autres limites, concernant la flexibilité des règles pour la mise à jour des listes d'engagements par exemple. Ainsi, ces règles, quoiqu'implémentables et formulées de manière précise, ne modélisent pas toutes les situations de dialogue imaginables (possibles). Ainsi, par exemple, la règle 1 (dans le paragraphe 7.2.1) pour la mise à jour de la liste d'engagements d'un locuteur L en situation de monologue sous-génére les engagements de L . Pour voir cela, considérons le dialogue suivant, entre deux locuteurs U et L :

U_1 : Jean est entré dans la chambre π_1 .

L_1 : Il s'est allongé dans son lit π_2 .

U_2 : Il s'est mis à feuilleter un livre π_3 .

A la fin de ce dialogue, U s'est engagé non seulement au fait que Jean s'est allongé dans son lit, mais au fait que Jean a fait cela *dans la chambre, juste après être entré dans la chambre*. Ce contenu, écrit en italiques, est une conséquence du fait qu'après avoir produit L_1 , L s'est engagé à $\Sigma_{Narration(\pi_1, \pi_2)}$. Mais la règle 1 dans le paragraphe 7.2.1 prédit que, juste après que π_3 est produit, U ne s'engage qu'à $\Sigma_{Narration(\pi_2, \pi_3)}$. En réalité, U a besoin de s'engager à $\Sigma_{Narration(\pi_1, \pi_2)} \wedge \Sigma_{Narration(\pi_2, \pi_3)}$. Cependant, ce type de phénomène (de narration co-construite par les interlocuteurs) n'est pas, pour l'instant, modélisé de manière suffisante dans notre approche. Il n'en est pas moins vrai que pour les situations typiques de dialogues orientés service entre un client et un système (par exemple, bibliothécaire virtuel) ce type de conversation n'est pas particulièrement représentatif ou pertinent. Autrement dit, si la place de U avait été prise par une machine, elle n'aurait pas produit des tours de parole comme U_2 , parce que, dans ce cas-là, la machine n'aurait pas eu l'information que L s'était engagé à $K(\pi_1)$, par conséquent, elle n'aurait pas autorisé une ellipse dans π_3 , en préférant de produire un énoncé plus explicite, du type 'Ensuite, il s'est mis

à feuilleter un livre dans la chambre'. Il est vrai que cet énoncé est moins naturel que π_3 , mais cela ne nuit tout de même pas au succès du dialogue.

Une autre règle pas assez généralisable à toutes les situations de dialogue imaginables est la partie (a) de la règle 2 dans le paragraphe 7.2.1. En effet, considérons le dialogue suivant, entre deux locuteurs U et L :

U_1 : Je voudrais qu'on se rencontre dans les deux prochaines semaines ^{π_1} . Est-ce qu'un vendredi, ça t'ira ? ^{π_2}

L_1 : Je ne suis pas disponible le 15 ^{π_3} .

Suite à L_1 , L ne s'engage pas seulement à $\Sigma_{IQAP(\pi_2, \pi_3)}$, mais aussi à la contribution illocutoire de la question π_2 , déterminée par la relation $Q - \text{Elab}(\pi_1, \pi_2)$ (parce que c'est cet acte qui établit le fait que la question de U concerne les vendredis *dans les deux prochaines semaines*, en opposition avec les vendredis en général). En fait, cette règle permet à L de s'engager justement à $K(\pi_2)$, concernant un vendredi dans les deux prochaines semaines, car l'analyse sémantique et l'interprétation pragmatique du tour U_1 auraient ancré ce vendredi de manière appropriée. Par conséquent, il n'est pas nécessaire que L s'engage à $\Sigma_{Q-\text{Elab}(\pi_1, \pi_2)}$; cette relation ne sert qu'à ancrer le référent 'vendredi' ; ainsi, le manque de généralité de la règle 2.(a) ne nuit pas, lui non plus, au succès du dialogue. Cela parce qu'il faut le rappeler, le but du cadre proposé ici n'est pas d'assurer une interprétation du dialogue aussi complète et fidèle que possible aux représentations humaines, mais juste d'offrir un mécanisme calculable pour gérer avec succès un ensemble limité de dialogues, orientés service.

Une autre critique concerne la manière dont nous définissons la négation d'une question ; on pourrait mettre en avant l'objection comme quoi un énoncé interrogatif est du mauvais type sémantique pour l'opération de négation. C'est pourquoi nous avons besoin de clarifier ce point. Considérons par exemple une question du type 'Est-ce que ce livre vous convient ?' ^{π} . Au niveau sémantique, un tel énoncé se représenterait au niveau logique (par la fonction $K(\pi)$) par un prédicat qui contiendrait une variable non-initialisée :

$\exists Y, Z : \text{object}('book') \wedge \text{feature}('book', Y) \wedge \text{title}(Y) \wedge \text{equals}(Y, '<book_title>') \wedge \text{want}(\neg\text{emitter}(\pi), 'book', Z) \wedge \text{equals}(Z, '?')$.

Ici, la variable non-initialisée est le booléen Z qui contient la valeur de vérité du prédicat $\text{want}/3$, qui est vrai si l'entité désignée par son premier argument (dans notre cas, le destinataire de π , $\neg\text{emitter}(\pi)$) désire l'entité désignée par le deuxième argument (dans notre cas, le livre 'book', dont le titre est donné par la valeur de la variable Y). Cela est cohérent avec notre définition des questions (cf. chapitre 3). La négation d'une telle question ne consiste pas à nier, de manière classique, chaque prédicat dans la conjonction et ensuite remplacer la conjonction par une disjonction, mais à attribuer la valeur 0 au booléen Z ; donc, dans notre cas, $\neg K(\pi)$ a la même forme que $K(\pi)$, à part le dernier prédicat, qui a la forme $\dots\text{equals}(Z, 0)$.

Enfin, une autre limite (réelle cette fois-ci) de notre cadre concerne l'incapacité de traiter de manière satisfaisante des dialogues où un interlocuteur met en cause non pas la vérité des énoncés, mais le lien *rhétorique* (et éventuellement causal) entre eux. Le problème avec notre cadre est qu'il a tendance à enlever trop de choses des listes d'engagements, en n'arrivant ainsi pas à saisir la différence subtile entre la conteste du contenu d'un énoncé et la conteste d'une relation rhétorique entre deux énoncés. Pour mieux comprendre le problème, nous considérons un dialogue entre deux locuteurs M et U (M pourrait être modélisé par un ordinateur) :

M_1 : Ce livre vient d'être interdit ^{π_1} . On y dévoilait des secrets d'état ^{π_2} .

U_1 : Non ! Il y avait trop de choses obscènes ^{π_3} .

M_2 : Ah, d'accord ^{π_4} .

Après le tour de parole M_2 , le locuteur M devrait être engagé à $\Sigma_{Consequence(\pi_3, \pi_1)}$ ⁹. Mais U réalise par la suite un mouvement correctif qui ne met pas que π_2 en question, mais aussi π : $Consequence(\pi_2, \pi_1)$ du premier tour de M (c'est-à-dire, à la fin de U_1 , U s'engage à $\Sigma_{P-Corr(\pi, \pi_3)} \wedge \Sigma_{P-Corr(\pi_2, \pi_3)}$, où $\pi \equiv Consequence(\pi_2, \pi_1)$). Par conséquent, nos règles enlèvent π (et par conséquent π_1 et π_2) de la liste d'engagements de M et, par la suite, les engagements à $\Sigma_{P-Corr(\pi, \pi_3)}$ et à $\Sigma_{P-Corr(\pi_2, \pi_3)}$ sont perdus. M ne reste plus engagé qu'à $K(\pi_3)$ et, éventuellement, à $K(\pi_1)$, si nous supposons que M s'est aperçu du mouvement de $P - Corr(\pi_2, \pi_3)$ dans U_1 . Donc, au meilleur des cas, nous perdons l'engagement de M à $\Sigma_{P-Corr(Consequence(\pi_2, \pi_1), \pi_3)}$.

En dépit de toutes ces critiques, il n'en reste pas moins que le contrôle du degré de force illocutoire en dialogues à plusieurs locuteurs est particulièrement utile dans des applications qui vont même au delà des dialogues orientés service ; nous pouvons considérer par exemple la narration interactive [171], ou les jeux sur ordinateur où des interactions linguistiques sont nécessaires. Cependant, pour pouvoir déployer ce cadre dans une application réelle, il faudrait y avoir un moyen de mettre en correspondance, de manière rigoureuse et programmatique, les différents degrés de force illocutoire et les formes de surface des énoncés. Ce sujet reste ouvert à des recherches futures, mais nous présentons cependant dans l'annexe A un ensemble de démarches préliminaires dans ce sens.

⁹Pour ne pas compliquer inutilement le traitement, nous supposons que tout le tour de parole U_1 constitue un seul énoncé, π_3 .

Chapitre 8

Vers la génération de surface : choix des connecteurs discursifs

8.1 Introduction

Les connecteurs discursifs sont importants pour guider la compréhension des messages¹ que les locuteurs transmettent sous forme linguistique en dialoguant. Par conséquent, ils ont reçu relativement beaucoup d'attention dans les recherches en linguistique, surtout lorsque l'interprétation des énoncés est concernée [55], [118]. En traitement automatique des langues, les connecteurs sont regardés en général de deux manières différentes, selon les buts assumés :

- en analyse morpho-syntaxique et sémantique, les connecteurs ont souvent un rôle modéré, car, d'un côté ils subissent et imposent peu de contraintes aux niveaux morphologique et syntaxique (voir par exemple, en français, l'emploi du subjonctif du verbe, après *bien que*) et, d'un autre côté, ils n'apportent rien à la sémantique des énoncés, considérée de manière vériconditionnelle et / ou isolée [89]² ;
- en interprétation / analyse du discours (qui remonte surtout à la prise en compte des aspects pragmatiques de la langue), les connecteurs ont une place importante, car leur présence guide le processus d'analyse [107].

En génération du langage naturel, les connecteurs sont en général considérés comme les manifestations superficielles (c'est-à-dire, au niveau de la réalisation de surface) des relations rhétoriques qui relient des énoncés distincts pour constituer des discours [147] ; par exemple, le connecteur *mais* matérialise une relation de *correction* par rapport à ce qui avait été dit auparavant. De toutes façons, cette approche s'avère souvent insuffisante, car, comme l'a montré Moeschler [118], les **conditions d'emploi** de chaque connecteur impliquent souvent plusieurs énoncés, reliés de manières assez diverses.

Au niveau computationnel, les propos de Moeschler doivent être développés sous forme d'algorithmes, où les entrées et les sorties soient clairement précisées. A partir de ce constat, ce

¹Le mot « message » est pris ici dans un sens élargi, intégrant la portée spécifiée par la théorie de l'information et les visées illocutoires des sujets parlants.

²Il y a toutefois des exceptions à cette manière de considérer les connecteurs en analyse et en génération, comme par exemple la théorie « sens-texte » de Mel'čuk [147].

chapitre propose une formalisation computationnelle du choix de quelques connecteurs discursifs dans le cadre d'un module générique, conçu pour le contrôle pragmatique de la génération du langage en dialogue oral homme-machine [136].

Au niveau théorique, la nouveauté de l'approche proposée dans ce chapitre réside dans le fait qu'à la différence de Moeschler, qui, étant donné le connecteur, déduisait le schéma sémantico-pragmatique le plus pertinent et donc l'emploi du connecteur (causal ou argumentatif) [118], nous proposons de déduire le schéma sémantico-pragmatique le plus pertinent à partir de la structure rhétorique (SDRS) du dialogue et, par conséquent, de choisir le connecteur qui satisfasse le mieux ce schéma ; ainsi, la cohérence du discours sera également prise en compte.

Du point de vue applicatif, une telle démarche est utile lorsqu'il s'agit, pour la machine, de produire des répliques, à partir d'expressions figées à trous, ce qui est une approche courante en dialogue homme-machine finalisé par la tâche, étant données la facilité de l'implémentation et la vitesse d'exécution ; les expressions figées sont ensuite collées *via* les connecteurs calculés. Nous générons donc un phénomène de surface (les connecteurs discursifs) de manière motivée du point de vue pragmatique et sémantique.

Pour un état de l'art concis sur ce sujet, on peut retenir, d'un côté, les démarches convergentes avec celles de Moeschler, qui concernent seulement l'*interprétation* des connecteurs du discours ou, d'un autre côté, l'interprétation des connecteurs surtout en termes de relations de discours, soit dans la direction fixée par Marcu [108], soit selon l'approche de Lascarides en SDRT [160]. En ce qui concerne le choix des connecteurs en génération, des travaux représentatifs ont été poursuivis par Marcu, qui reliait les connecteurs à des relations discursives seulement [108] ; donc, selon la terminologie de Moeschler, seul le « cotexte » était pris en compte, en opposition avec le « contexte » sémantico-pragmatique, qui légitime ou bloque la production d'un connecteur, surtout lorsque le dialogue oral est concerné.

Ici, à partir de la structure rhétorique (cotexte), nous calculons un contexte sémantico-pragmatique, que nous filtrons ensuite à travers le cotexte du dialogue entier, et qui pilote le choix des connecteurs. Ces connecteurs sont d'au moins trois types :

- (i) connecteurs **concessifs**, qui marquent une forme d'opposition ou de contraste entre les énoncés reliés ; exemples : *mais, quand même, pourtant* ou *bien que* ;
- (ii) connecteurs **argumentatifs**, qui marquent une forme de conséquence ou conditionnement entre les énoncés reliés ; exemples : *parce que, puisque, car* ou *donc* ;
- (iii) connecteurs **phatiques**, qui marquent surtout des aspects expressifs des énoncés ; exemples : *donc* utilisé comme tic verbal, ou *eh ben*.

Dans ce chapitre, seuls les connecteurs concessifs seront analysés de manière détaillée, car cela nous permet, nous croyons, de mieux séparer les aspects sémantiques (du type « conséquence logique » des contraintes rhétoriques sur la réalisation de ces connecteurs). Ainsi, la section suivante constitue le cœur du chapitre et présente les sémantiques des connecteurs pragmatiques considérés, aussi que les détails formels et computationnels du mécanisme pilotant le choix des connecteurs en génération ; ce mécanisme sera ensuite évalué de manière qualitative *via* un exemple typique de dialogue. La troisième section formalise des éléments de la Théorie de l'argumentation de Anscombe et Ducrot [6], pour proposer un mécanisme d'ordonnement argumentatif des énoncés en dialogue ; cela est utile notamment pour générer des connecteurs argumentatifs, tels que *parce que* ou *car*, soit des connecteurs plurisémantiques, comme *mais* ;

TAB. 8.1 – Connecteurs concessifs et relations rhétoriques, en RST

Connecteur (français)	Connecteur (anglais)	Relation rhétorique (RST)
<i>mais</i>	<i>but</i>	CONTRAST
<i>quand même</i>	<i>though, anyway</i>	CONCESSION, CONTRAST
<i>pourtant</i>	<i>yet</i>	ANTITHESIS
<i>bien que</i>	<i>although</i>	CONCESSION, ELABORATION

cependant, seul ce dernier sera analysé de manière détaillée dans ce chapitre. La quatrième section conclut le chapitre et présente des pistes pour continuer cette recherche.

8.2 Génération des connecteurs concessifs en dialogue

8.2.1 Relations rhétoriques et connecteurs discursifs

Un des travaux les plus représentatifs lorsqu'on parle de l'interprétation rhétorique des textes à partir des connecteurs discursifs est celui de Marcu [108] où, dans le cadre de la RST (« Rhetorical Structure Theory »), il propose un ensemble de méthodes pour calculer l'arbre de discours pour des textes monologiques. Dans cette tâche, ce travail s'appuie sur des correspondances entre connecteurs discursifs (plus de 450 connecteurs différents) et relations rhétoriques (54 relations rhétoriques en RST) ; ces correspondances sont à leur tour dérivées d'un corpus d'environ 7600 textes tirés du corpus Brown (en langue anglaise), en moyenne 17 textes par connecteur considéré [108].

Ainsi, suite à l'analyse de ce corpus l'auteur a trouvé des associations comme celles présentées dans le tableau 8.1, où nous avons indiqué le connecteur français, sa traduction anglaise et les relations rhétoriques lui correspondant, en RST.

Même si Marcu propose un ensemble de contraintes structurelles aux arbres RST correspondant aux discours monologiques, il ne donne pas de consignes sémantico-pragmatiques sur les connecteurs de discours en tant que tels ; il propose seulement des contraintes « syntaxiques » sur les relations de discours, pour les associer ensuite, à partir d'un corpus de textes, aux connecteurs. Cela implique l'absence d'une sémantique enrichie des connecteurs, donc le contexte d'utilisation des connecteurs est réduit au *cotexte* formé de deux énoncés (ceux reliés par le connecteurs) et de la relation rhétorique trouvée valide entre ces énoncés. Même si une telle approche pouvait paraître légitime pour l'analyse des textes, lorsqu'il s'agit de motiver la *génération* par rapport au contexte « pragmatique » (surtout en dialogue), elle s'avère insuffisante, car parfois il n'y a pas moyen de choisir entre deux connecteurs différents, étant donné une relation de discours ; par exemple, le choix entre *bien que* et *quand même* n'est pas évident, lorsqu'on part de la relation rhétorique de CONCESSION.

Ce type d'ambiguïté a été constaté également en SDRT ; par exemple l'étude de Sporelander et Lascarides [160] sur la dérivation des relations rhétoriques en SDRT à partir des marqueurs de surface a démontré le manque de fiabilité des correspondances entre les marques textuelles de discours (connecteurs pragmatiques, points d'exclamation, d'interrogation, etc.) et les relations de discours : lorsqu'on essayait de calculer la structure de discours d'un texte à partir

des marques de surfaces, on obtenait une F -mesure voisinant les 55-60 %, par rapport à une annotation manuelle des mêmes discours.

8.2.2 Aperçus sémantico-pragmatiques des connecteurs

Les connecteurs discursifs marquant la **concession** ont été bien étudiés, surtout dans l'école genevoise de linguistique (Moeschler, Roulet, Rubattel, etc.) [118] ; ces travaux ont abouti à des modélisations sémantiques formelles des connecteurs tels que *mais*, *quand même*, *pourtant* ou *bien que*. Dans ce paragraphe nous donnons un bref aperçu d'une telle modélisation, à partir de la présentation de [118].

Pour cela, nous considérons d'abord un ensemble de notations (dont une partie a déjà été introduite dans les chapitres antérieurs), utilisées tout au long de ce chapitre :

- α, β, \dots ::= étiquettes des énoncés dans le discours ;
- $*$::= marque de connexion entre deux énoncés du discours ;
- TRUE, FALSE ::= les valeurs de vérité « vrai », respectivement « faux » ;
- $\text{equals}(\alpha, \varphi)$::= prédicat binaire vrai si et seulement si la variable α est égale (c'est-à-dire, unifiée) avec la valeur φ ;
- $\Rightarrow, \neg, \wedge, \vee$::= connecteurs logiques ayant leur signification classique en logique [63] ;
- $\in, \exists, \subset, \supset$::= connecteurs ayant leur signification classique en théorie des ensembles ;
- \exists, \forall ::= quantificateurs logiques existentiel, respectivement universel ;
- $K(\alpha)$::= fonction retournant la sémantique (en logique du premier ordre) de l'énoncé α ;
- $\nu(\lambda)$::= fonction retournant la valeur de vérité de la formule logique λ ; donc, désignant par Λ l'ensemble des formules logiques définissables pour un alphabet donné, nous avons que $\nu : \Lambda \rightarrow \{\text{TRUE}, \text{FALSE}\}$;
- $\text{cause}(\alpha, \beta)$::= prédicat binaire vrai si et seulement si la modification de contexte du discours, déterminée par l'énoncé α , est une *cause* suffisante pour l'ajout du contenu de l'énoncé β au contexte ;
- $\rightarrow (\alpha, \beta)$::= relation argumentative entre les énoncés α et β , vraie si et seulement si un locuteur utilise l'énoncé α pour faire admettre le contenu propositionnel où les visées illocutoires de l'énoncé β [55] ;
- $<$::= relation de *force argumentative* entre deux énoncés : $\alpha < \beta$ si et seulement si β est plus *pertinent* pour son émetteur (qui tente d'argumenter le contenu d'un autre énoncé, soit-il γ) que α [118] ;
- $\nu(\alpha, \beta)$::= relation de *norme*, vraie si et seulement si *d'ordinaire* α implique β ; la norme est donc une implication révisable selon le contexte ;
- $\text{smaller}(\phi, \psi)$::= prédicat binaire vrai si et seulement si la valeur ϕ est inférieure à la valeur ψ ;
- $\text{ClassOf}(\phi, \Phi)$::= prédicat binaire vrai si et seulement si l'élément ϕ appartient à l'ensemble Φ ;
- $\bar{\vee}$::= opération logique de « ou exclusif ».

Ainsi, [118] définit deux types de concession, **causale** et **argumentative** qui se particularisent pour chacun des connecteurs considérés – *mais*, *quand même*, *pourtant* ou *bien que*. Par conséquent, [118] propose un ensemble de sémantiques pour ces quatre connecteurs, dont nous

ne retenons, de toute façon, que les aspects « procéduraux », car le but assumé dans ce chapitre est de proposer un cadre *opérationnel* pour la génération de ces connecteurs en dialogue homme-machine.

Ainsi, *mais* et *quand même* se définissent à travers un « carré argumentatif » (délimité par trois énoncés, α , β et γ), formalisable de manière concise comme ci-dessous :

1. *mais* peut être employé de deux manières, désignées par *mais*₁ (pour l'emploi de **relation directe**) et *mais*₂ (pour l'emploi de **relation indirecte**) :

$$- \alpha \text{ mais}_1 \neg\gamma ::= \exists\beta : \rightarrow (\alpha, \gamma) \wedge (\gamma \bar{\vee} \neg\gamma) \wedge (\alpha < \beta) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE}) \wedge \rightarrow (\beta, \neg\gamma) ;$$

exemple : Il fait beau *mais*₁ je ne sors pas . – α est 'il fait beau', γ est 'je sors' et β est par exemple 'je ne me sens pas bien'.

$$- \alpha \text{ mais}_2 \beta ::= \exists\gamma : \rightarrow (\alpha, \gamma) \wedge (\gamma \bar{\vee} \neg\gamma) \wedge \rightarrow (\beta, \neg\gamma) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE}) ;$$

exemple : Il fait beau *mais*₂ j'ai mal à la tête. – α est 'il fait beau', β est 'j'ai mal à la tête' et γ est par exemple 'je sors'.

2. *quand même* peut être employé en relation directe seulement, ainsi :

$$\alpha \text{ quand même } \neg\gamma ::= \exists\beta : v(\alpha, \gamma) \wedge (\gamma \bar{\vee} \neg\gamma) \wedge \rightarrow (\beta, \neg\gamma) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE}) ;$$

exemple : Il fait beau. Je reste *quand même* chez moi. – α est 'il fait beau', β est par exemple 'j'ai mal à la tête' et γ est 'je sors' .

D'un autre côté, les connecteurs *pourtant* et *bien que* se définissent à travers un « triangle causal » (délimité par deux énoncés, α et β), formalisable ainsi :

1. *pourtant* peut être employé de deux manières, selon les deux permutations possibles du triangle causal [118] :

$$- \alpha \text{ pourtant}_1 \beta ::= \text{cause}(\alpha, \neg\beta) \wedge (\beta \bar{\vee} \neg\beta) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE}) ;$$

exemple : Il pleut, *pourtant*₁ je me balade dans le parc. – α est 'il pleut' et β , 'je me balade...'

$$- \alpha \text{ pourtant}_2 \beta ::= \text{cause}(\beta, \neg\alpha) \wedge (\alpha \bar{\vee} \neg\alpha) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE}) ;$$

exemple : Je suis gentil avec lui, *pourtant*₂ il me dit de gros mots toujours. – α est 'je suis gentil...' et β , 'il me dit de gros mots...'.

2. *bien que* peut également être employé de deux manières, selon les permutations du triangle causal :

$$- \text{bien que}_1 \alpha, \beta ::= \text{cause}(\alpha, \neg\beta) \wedge (\beta \bar{\vee} \neg\beta) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE}) ;$$

exemple : *Bien qu'*₁ il fasse mauvais, je sors. – α est 'il fait mauvais' et β , 'je sors'.

$$- \alpha \text{ bien que}_2 \beta ::= \text{cause}(\beta, \neg\alpha) \wedge (\alpha \bar{\vee} \neg\alpha) \wedge \text{equals}(v(K(\alpha)), \text{TRUE}) \wedge \text{equals}(v(K(\beta)), \text{TRUE}) ;$$

exemple : Je me balade dans le parc, *bien qu'*₂ il pleuve. – α est 'je me balade...' et β , 'il pleut'.

Dans ce paragraphe nous avons présenté les sémantiques de quatre connecteurs concessifs, *mais*, *quand même*, *pourtant* et *bien que*, formalisées dans une approche qui s'inspire des travaux de l'école genevoise [118]. Pour une spécification algorithmique du processus de génération de ces connecteurs, à partir de leurs sémantiques, il y a au moins un point à améliorer dans les sémantiques présentées : nous observons qu'au niveau formel il n'y a aucune différence entre les sémantiques de *pourtant* et *bien que*. Comment savoir alors quel connecteur produire, dans un contexte donné ? Pour l'interprétation, Moeschler précise seulement que *bien que* stipule une plus forte détermination causale entre ses arguments, par rapport à *pourtant* [118] ; donc, il faudra exprimer cette idée de manière formelle, en tant que règle ou contrainte supplémentaire agissant sur la sémantique de ce connecteur.

8.2.3 Structure de discours et sémantique des connecteurs

L'idée fondamentale de ce chapitre est de « calculer » le connecteur le plus approprié en génération dialogique, par rapport au contexte sémantico-pragmatique de la conversation. Ainsi, pour cela nous partons d'une représentation profonde du cotexte associé au dialogue, à laquelle nous ajouterons des contraintes contextuelles.

Nous supposons que le point de départ pour le calcul des connecteurs est une structure segmentée de discours (SDRS), calculé en préalable par un module de structuration rhétorique (selon les principes présentés dans les chapitres 3, 4 et 5 de cette thèse).

Pour que nous puissions utiliser cette structure rhétorique dans le calcul des connecteurs, nous spécifions une grammaire pour la représenter, utilisant les informations fournies par le module de structuration rhétorique. Une telle grammaire suppose que les informations sur la SDRS consistent en (i) le nombre D des énoncés connectés par des relations rhétoriques, (ii) le nombre R des relations rhétoriques qui relient les énoncés, (iii) l'ensemble Π d'étiquettes π des énoncés, (iv) l'ensemble P des étiquettes des relations rhétoriques³, (v) les formes logiques K des énoncés dans la structure de discours, (vi) les sémantiques (exprimées dans une logique du premier ordre – voir chapitre 3 dans cette thèse) Σ des relations rhétoriques dans la structure de discours. Cette grammaire est présentée de manière détaillée ci-dessous :

- | | |
|--|--|
| 1. $SDRS \rightarrow (D, R, \Pi, P)$; | 5. $\rho_j : j \in \{1, \dots, R\} \rightarrow \rho_j(\pi_i, \pi_k) : j \in \{1, \dots, R\}; i, k \in \{1, \dots, D\} \rho_j(\pi_i, \rho_k) : i \in \{1, \dots, D\}; j, k \in \{1, \dots, R\}$; |
| 2. $\Pi \rightarrow \{\pi_1, \dots, \pi_D\}$; | |
| 3. $\pi_i : i \in \{1, \dots, D\} \rightarrow K_i : i \in \{1, \dots, D\}$; | |
| 4. $P \rightarrow \{\rho_1, \dots, \rho_R\}$; | 6. $\rho_j : j \in \{1, \dots, R\} \rightarrow \Sigma_j : j \in \{1, \dots, R\}$. |

Pour relier cette structure rhétorique aux spécifications sémantiques des connecteurs concessifs (présentées dans le paragraphe antérieur), nous allons transposer le carré argumentatif et le triangle causal qui situent les quatre connecteurs considérés, dans des contraintes de type rhétorique ; pour cela, nous nous appuyons sur les sémantiques des relations rhétoriques, en tant que précisées dans le chapitre 3 de cette thèse. Ainsi, nous proposons tout d'abord un ensemble de correspondances entre les relations sémantiques spécifiant les connecteurs et les relations rhé-

³Les étiquettes des relations rhétoriques sont identiques à leurs noms [11].

TAB. 8.2 – Relations sémantiques (Moeschler) et relations rhétoriques (SDRT)

Relation sémantique	Relation rhétorique
$\bar{\vee}(\alpha, \beta)$	$Contrast(\alpha, \beta) \wedge (\alpha \vee \beta)$
$cause(\alpha, \beta)$	$\alpha \Rightarrow \beta$
$\rightarrow(\alpha, \beta)$	$Consequence(\alpha, \beta)$
$\nu(\alpha, \beta)$	$Background(\alpha, \beta)$

toriques ; ces correspondances, présentées dans le tableau 8.2 sont motivées également par les définitions de Moeschler [118] et par les spécifications de la SDRT [11].

Nous appuyant sur ces correspondances, nous pourrions spécifier les sémantiques des connecteurs, en termes des relations de discours (et des connecteurs logiques usuels). Cependant, une observation préalable s'impose : dans le paragraphe précédent il n'y avait pas de distinction sémantique formelle entre *pourtant* et *bien que* ; de toutes façons, l'idée d'une plus forte détermination causale agencée par *bien que* peut être formalisée via une contrainte supplémentaire sur ce connecteur : *que le temps de l'action décrite par l'énoncé qui succède ce connecteur précède le temps de l'action décrite par le deuxième énoncé sous l'incidence de ce connecteur*. Au niveau logique cela sera exprimé via la fonction $\Delta(\alpha)$ qui, nous le rappelons (voir chapitre 3 pour une définition complète de cette fonction), précise (retourne) le temps de l'action décrite par l'énoncé libellé α .

Ainsi, les sémantiques des connecteurs concessifs considérés se formalisent, en termes rhétoriques, comme ci-dessous :

1. α *mais*₁ $\neg\gamma ::= \exists\beta : Consequence(\alpha, \gamma) \wedge Contrast(\gamma, \neg\gamma) \wedge Consequence(\beta, \neg\gamma) \wedge equals(\nu(K(\alpha)), TRUE) \wedge equals(\nu(K(\beta)), TRUE) \wedge equals(\nu(K(\gamma)), FALSE) \wedge <(\alpha, \beta)$;
2. α *mais*₂ $\beta ::= \exists\gamma : Consequence(\alpha, \gamma) \wedge Contrast(\gamma, \neg\gamma) \wedge Consequence(\beta, \neg\gamma) \wedge equals(\nu(K(\alpha)), TRUE) \wedge equals(\nu(K(\beta)), TRUE) \wedge equals(\nu(K(\gamma)), FALSE)$;
3. α *quand même* $\neg\gamma ::= \exists\beta : Background(\alpha, \gamma) \wedge Contrast(\gamma, \neg\gamma) \wedge Consequence(\beta, \neg\gamma) \wedge equals(\nu(K(\alpha)), TRUE) \wedge equals(\nu(K(\beta)), TRUE) \wedge equals(\nu(K(\gamma)), FALSE)$;
4. α *pourtant*₁ $\beta ::= (\alpha \Rightarrow \neg\beta) \wedge Contrast(\beta, \neg\beta) \wedge equals(\nu(K(\alpha)), TRUE) \wedge equals(\nu(K(\beta)), TRUE)$;
5. α *pourtant*₂ $\beta ::= (\beta \Rightarrow \neg\alpha) \wedge Contrast(\alpha, \neg\alpha) \wedge equals(\nu(K(\alpha)), TRUE) \wedge equals(\nu(K(\beta)), TRUE)$;
6. α *bien que*₁ $\alpha, \beta ::= (\alpha \Rightarrow \neg\beta) \wedge Contrast(\beta, \neg\beta) \wedge equals(\nu(K(\alpha)), TRUE) \wedge equals(\nu(K(\beta)), TRUE) \wedge smaller(\Delta t(\alpha), \Delta t(\beta))$;
7. α *bien que*₂ $\beta ::= (\beta \Rightarrow \neg\alpha) \wedge Contrast(\alpha, \neg\alpha) \wedge equals(\nu(K(\alpha)), TRUE) \wedge equals(\nu(K(\beta)), TRUE) \wedge smaller(\Delta t(\beta), \Delta t(\alpha))$.

Dans ces sémantiques, les énoncés non reliés par les connecteurs, mais intervenant dans les contraintes sur ces derniers (γ ou β , selon le cas) sont tirés soit du cotexte, soit du contexte :

- si l'énoncé en question est présent dans le discours, dans des tours de parole antérieurs, nous nous appuyons sur lui ;

- sinon, nous déduisons cet énoncé à partir des énoncés antérieurs et de l'**ontologie de tâches** [35] (le contexte), *via* un processus inférentiel.

Une dernière remarque à l'égard des sémantiques des connecteurs concessifs (et du lien entre celles-ci et la structure rhétorique) concerne la relation « < /2 », de force argumentative, non formalisée par Moeschler ou dans les sections antérieures de ce chapitre. Ainsi, nous nous appuyons de manière explicite sur le contexte du *dialogue*, posant qu'il peut y avoir deux interlocuteurs, l'émetteur d'un énoncé courant, désigné comme ci-dessus, par la fonction `emitter/1` et l'allocutaire, désigné par `¬emitter/1` ; par conséquent, il n'y a pas d'équivalence entre des expressions du type $*(\neg\text{emitter}(\alpha), \phi)$ et $\neg *(\text{emitter}(\alpha), \phi)$, où « * » désigne un prédicat binaire quelconque et « ϕ », un terme quelconque. Ceci étant, nous posons comme hypothèse de travail qu'un énoncé dont le contenu fait une référence explicite à son émetteur a une force argumentative plus élevée par rapport à un énoncé qui ne le fait pas, ou qu'un énoncé qui fait une référence explicite à l'allocutaire (donc, au destinataire) a une force argumentative plus élevée par rapport à un énoncé dont le contenu ne fait référence ni à cet allocutaire, ni à l'émetteur. Précisant que deux énoncés dont la force argumentative est à comparer ont le même émetteur, ces consignes peuvent s'exprimer sous forme logique par :

$$\alpha < \beta ::= (\text{ClassOf}(\text{emitter}(\beta), K(\beta)) \wedge \neg \text{ClassOf}(\text{emitter}(\alpha), K(\alpha))) \vee (\text{ClassOf}(\neg\text{emitter}(\beta), K(\beta)) \wedge \neg \text{ClassOf}(\neg\text{emitter}(\alpha), K(\alpha)) \wedge \neg \text{ClassOf}(\text{emitter}(\alpha), K(\alpha))) \wedge \text{equals}(\text{emitter}(\alpha), \text{emitter}(\beta)) \wedge \text{equals}(\neg\text{emitter}(\alpha), \neg\text{emitter}(\beta)).$$

8.2.4 Choix des connecteurs : algorithme

Le cadre posé dans le paragraphe précédent pour l'expression sémantique, amendée par la structure rhétorique, des connecteurs concessifs en dialogue sera mis en valeur *via* une procédure conçue pour générer les connecteurs appropriés, lorsqu'il s'agit, pour la machine, de produire un énoncé dialogique.

Les entrées dans cette procédure consistent en (i) la structure rhétorique du dialogue « courant » (c'est-à-dire, intégrant aussi la sémantique de la réplique censée être générée par la machine et donc, éventuellement, contenir un ou plusieurs connecteurs concessifs), (ii) les sémantiques des connecteurs (spécifiées sous forme de règles) et (iii) l'ontologie de tâches pour les dialogues (spécifiée sous forme de hiérarchie de concepts [35]).

La sortie de cette procédure consiste dans un ensemble de connecteurs qui peuvent relier les énoncés dans la réplique courante de la machine ; cette liste peut éventuellement être vide, mais dans le cas contraire, chaque entrée dans cette liste a trois éléments : (i) le connecteur choisi, (ii) son « argument gauche » et (iii), son argument droit ; donc une entrée type est de la forme : $(*, \pi_i, \pi_j)$, ce qui correspondra à une forme de surface du type $\pi_i * \pi_j$, où * est le connecteur et π_i et π_j , les libellés de deux énoncés. Donc, la procédure est spécifiée ci-dessous :

1. initialiser la liste des connecteurs : $\Lambda \leftarrow \emptyset$;
2. lire la structure de discours courante, $S(D, R, P, \Pi)$ (en vérifiant sa conformité à la grammaire des SDRS) ;
3. lire les étiquettes des énoncés à être générés par la machine, $\langle \pi \rangle$, avec $\langle \pi \rangle \subset \Pi$ et $|\langle \pi \rangle| < D$;

4. trier l'ensemble $\langle \pi \rangle$ par rapport à l'ordre d'apparition des énoncés (ordre spécifié par le composant de structuration rhétorique – voir notamment les chapitres 3 et 4 dans cette thèse);
5. **pour** chaque énoncé π_- dans $\langle \pi \rangle$:
 - (a) repérer les relations rhétoriques connectant l'énoncé π_- à des énoncés dans S ; soient ces relations spécifiées par l'ensemble $\langle \rho \rangle_{\pi_-} \subset P$, avec $|\langle \rho \rangle_{\pi_-}| \subset R$;
 - (b) repérer les arguments *gauches* des relations dans l'ensemble $\langle \rho \rangle_{\pi_-}$; soient ces arguments spécifiés par l'ensemble $\langle \gamma \rangle_{\pi_-} \subset \Pi$, avec $|\langle \gamma \rangle_{\pi_-}| \subset D$;
 - (c) **pour** chaque sémantique ξ des connecteurs ($\xi \in \Xi$, avec $|\Xi| = 7$, car il y a 7 connecteurs possibles) :
 - i. **appairier**($\xi, \pi_-, (\langle \rho \rangle_{\pi_-}, \langle \gamma \rangle_{\pi_-})$) ;
 - ii. **si** OK, **alors** $\Lambda \leftarrow \Lambda \cup (\xi, \pi^{\xi}, \pi_-)$;
 - iii. **sinon**, **alors** $\Lambda \leftarrow \lambda \cup \emptyset$;
6. **si** $|\langle \pi^{\xi}, \pi_- \rangle| > 1$ (la procédure **appairier** retourne plusieurs connecteurs entre deux énoncés), **alors** :
 - (a) **s'il** existe un connecteur qui a un maximum de contraintes, **alors** garder celui-ci et enlever le reste (par exemple, entre *pourtant* et *bien que*, garder le dernier et enlever le premier) ;
 - (b) **sinon**, **s'il** existe un connecteur qui assure un nombre maximal de relations rhétoriques, c'est-à-dire, qui contient un nombre maximal de relations rhétoriques *différentes* dans ses contraintes, **alors** :
 - i. **si** les sémantiques des connecteurs ne mènent pas à une contradiction logique, alors garder tous ces connecteurs ;
 - ii. **sinon**, **alors** garder celui-ci et enlever le reste (par exemple, entre *mais* et *quand même*, garder le dernier et enlever le premier, car celui-ci contient deux relations rhétoriques différentes, *Consequence* et *Contrast*, tandis que le dernier en contient trois - les deux qui viennent d'être précisées, auxquelles *Background* s'ajoute) ;
 - (c) **sinon**, **alors** choisir de manière aléatoire un des connecteurs (cette situation est peu probable, étant données les sémantiques des connecteurs et leur utilisation) ;
7. **retourner** Λ .

Le noyau de l'algorithme consiste dans la procédure **appairier/3**, qui prend en entrée la sémantique d'un connecteur, (l'étiquette d') un énoncé qui constituera l'argument *droit* du connecteur et une structure formée de deux ensembles : un ensemble de relations rhétoriques (censées avoir π_- en tant qu'argument droit) et, respectivement, un ensemble d'énoncés (censés être les arguments gauches des relations rhétoriques). La sortie de la procédure **appairier** est constituée d'une structure qui contient la sémantique d'un connecteur, son argument gauche et, respectivement, son argument droit (qui est censé être justement l'énoncé π_- lu en entrée). Ainsi, la procédure **appairier** opère de la manière suivante :

1. initialiser trois listes : $\bar{\Pi} \leftarrow \emptyset$, $\bar{P} \leftarrow \emptyset$, $\bar{C} \leftarrow \emptyset$; ces listes contiendront, respectivement, des étiquettes d'énoncés, des étiquettes de relations rhétoriques, et des connecteurs logiques ;
2. **pour** chaque terme dans la sémantique ξ du connecteur, le classer dans une des catégories suivantes : (i) étiquettes d'énoncés, (ii) relations de discours, (iii) connecteurs logiques ;
3. par conséquent, actualiser les listes $\bar{\Pi}$, \bar{P} et \bar{C} ;
4. **si** $\bar{P} \subset \langle \rho \rangle_{\pi_-}$, **alors** :
 - (a) unifier les termes dans $\bar{\Pi}$ qui sont les arguments des termes en \bar{P} , avec les termes en $\langle \gamma \rangle_{\pi_-}$ qui sont les arguments des termes en $\bar{P} \cap \langle \rho \rangle_{\pi_-}$;
 - (b) **pour** chaque connecteur logique en \bar{C} , vérifier si celui-ci peut être placé entre des entités en $\langle \rho \rangle_{\pi_-} \cap \bar{P}$ et $\langle \gamma \rangle_{\pi_-} \cap \bar{\Pi}$, afin que :
 - la succession des termes et connecteurs soit identique à celle spécifiée en ξ , à part les unifications de l'étape 4.(a) ;
 - la forme logique résultée (en étendant les énoncés à leurs sémantiques, exprimées *via* des prédicats dans l'ontologie de tâche, selon les détails précisés dans le chapitre 3 de cette thèse) ne mène pas à l'ensemble vide (c'est-à-dire, à une contradiction) ;
 cette étape représente en fait la résolution de la formule logique obtenue de ξ , en substituant les énoncés par des énoncés dans la structure de discours en entrée, pourvu que nous préservions les arguments des relations rhétoriques ;
 - (c) **si** OK, **alors** retourner OK, (ξ, π^{ξ}, π_-) ;
 - (d) **sinon, alors** retourner KO ;
5. **sinon, alors** retourner KO.

Cet algorithme sera mis à l'épreuve sur un exemple typique de dialogue homme-machine à deux locuteurs dans le contexte d'une application de réservation et emprunt de livres dans une bibliothèque.

8.2.5 Traçage sur une situation de dialogue

Nous démontrons l'opération des algorithmes proposés sur un fragment de dialogue entre le système et un client, où la machine est une bibliothécaire virtuelle :

U : Excusez-moi ^{π_{11}} , pourrais-je emprunter ce bouquin ? ^{π_{12}}

M : Monsieur, vous ne pouvez emprunter que cinq livres à la fois ^{π_{21}} , *mais* je ferai *quand même* une exception cette fois-ci ^{π_{22}} .

U : Merci beaucoup ! ^{π_{31}} Alors, je vous donne ma carte ? ^{π_{32}}

M : Bon ^{π_{41}} , *bien que* vous ayez déjà gardé un livre il y a deux mois ^{π_{42}} , je vous en prête maintenant quatre ^{π_{43}} ; *pourtant*, vous n'avez pas l'air plus concerné en ce moment... ^{π_{44}}

Les énoncés de *M* sont disponibles sous forme logique, en tant que fournis par le contrôleur de dialogue (voir chapitre 1 de cette thèse, mais cf. aussi [35]), tandis que ceux de *U*, sous forme linguistique ; à partir de ces derniers, l'analyseur sémantique calcule, pour ces énoncés, les

formes logiques correspondantes, auxquelles l'interpréteur pragmatique attache les informations concernant le type d'acte de langage véhiculé par chaque contribution dialogique de l'utilisateur [35]. A partir de ces informations, le composant de structuration rhétorique dans le module de génération calcule la SDRS qui intègre les formes logiques des énoncés censés être réalisés sous forme linguistique par M (voir chapitres 3, 4 et 5 dans cette thèse). Ainsi, pour le fragment de dialogue considéré, la SDRS calculée est exprimée, sous forme synthétique, ci-dessous :

$$\begin{aligned} & \text{Elaboration}(\pi_{11}, \pi_{12}) \wedge P\text{-Corr}(\pi_{12}, \pi_{21}) \wedge \text{Consequence}(\pi_{21}, \neg\pi_{22}) \wedge \text{ACK}(\pi_{22}, \pi_{31}) \wedge \\ & \text{Consequence}(\pi_{31}, \pi_{32}) \wedge \text{ACK}(\pi_{32}, \pi_{41}) \wedge P\text{-Elab}(\pi_{22}, \pi_{43}) \wedge \text{Consequence}(\pi_{42}, \neg\pi_{43}) \wedge \\ & \text{Consequence}(\pi_{43}, \neg\pi_{44}) \wedge \text{Narration}(\pi_{42}, \pi_{44}). \end{aligned}$$

Les relations de discours ont été calculées entre des paires d'énoncés et même entre des paires dont un élément est un énoncé sous forme niée⁴ ; cela est utile pour faire la correspondance entre des fragments de cette SDRS et les sémantiques des connecteurs. Ainsi, pour la première réplique de la machine, entre les énoncés π_{21} et π_{22} la procédure *apparier* retourne les éléments (*mais*, π_{21}, π_{22}) et (*quand même*, π_{21}, π_{22}), car, pour *mais* nous retrouvons les contraintes $\text{Consequence}(\pi_{21}, \neg\pi_{22}) \wedge \text{Contrast}(\pi_{22}, \neg\pi_{22}) \wedge \text{equals}(\nu(K(\pi_{21})), \text{TRUE}) \wedge \text{equals}(\nu(K(\neg\pi_{21})), \text{FALSE})$ et l'énoncé « manquant » β est retrouvé en tant que 'vous ne pouvez pas emprunter ce livre' qui est déduit par *modus ponens* [63] à partir de π_{21} et du fait (connu par le système à travers son ontologie de tâche) que U a déjà emprunté cinq livres qu'il n'a pas encore rendus ; pour *quand même*, il est déduit *via* les mêmes contraintes que ci-dessus, auxquelles s'ajoute la contrainte $\text{Background}(\pi_{21}, \neg\pi_{22})$, déduite par *modus ponens* du fait que U a déjà emprunté cinq livres et que, par défaut, selon le règlement, la bibliothèque ne prête pas plus de cinq livres à la fois. Dans ce contexte, nous avons trois choix : (i) supprimer *mais*, (ii) supprimer *quand même*, (iii) garder les deux connecteurs ; selon l'algorithme précisé dans le paragraphe antérieur nous devrions adopter un des choix (ii) ou (iii). Du fait que (iii) ne mène à aucune contradiction logique, nous l'adoptons, car ainsi nous augmentons la pertinence *ilocutoire* de la contribution langagière de M . De toutes façons, pour pouvoir garder *mais* nous devons vérifier la satisfaction de $\langle \pi_{21}, \beta \rangle$, ce qui est vrai parce que π_{21} exprime une vérité générale – 'on peut emprunter au plus cinq livres à la fois', tandis que β exprime un fait relié à U , donc : $\text{ClassOf}(\neg\text{emitter}(\beta), K(\beta))$.

La deuxième contribution de la machine est traitée de manière similaire : entre les énoncés π_{42} et π_{43} la procédure *apparier* retourne (*bien que*, π_{42}, π_{43}) et (*pourtant*, π_{42}, π_{43}), car nous retrouvons la contrainte $\pi_{42} \Rightarrow \neg\pi_{43}$, déduite de la relation $\text{Consequence}(\pi_{42}, \neg\pi_{43})$ et du fait, présent dans l'ontologie de tâches, que si un client avait perdu un document, la bibliothèque serait devenue plus stricte avec celui-ci, ce qui n'est pas compatible avec la concession de prêter plus de documents que le règlement ne le prévoit ; toutefois, nous en retenons seulement *bien que*, car, selon l'algorithme de calcul, ce connecteur est défini par plus de contraintes : à partir de $K(\pi_{42})$ et de $K(\pi_{43})$ (les sémantiques des énoncés π_{43} et π_{44}) nous déduisons $\text{smaller}(\Delta t(\pi_{42}), \Delta t(\pi_{43}))$. En revanche, pour les énoncés π_{43} et π_{44} , la procédure *apparier*, opérant à l'instar du cas précédent, ne retourne quand même que *pourtant*, car, du fait que le temps de ces deux énoncés est le même (le présent), nous avons que $\text{equals}(\Delta t(\pi_{43}), \Delta t(\pi_{44}))$, donc les contraintes pour *bien que* ne sont pas satisfaites.

⁴Cela représente une nouveauté par rapport aux principes présentés dans le chapitre 3 de cette thèse.

8.3 Ordonnancement argumentatif des énoncés

La formalisation de la force argumentative proposée dans le paragraphe 8.2.3 est insuffisante, au sens où il n'est pas évident comment on pourrait l'étendre au cas du dialogue entre la machine et plusieurs locuteurs ; en effet, les interlocuteurs de la machine ne sont plus réduits à un seul allocataire (\neg emitter()); du fait que plusieurs interlocuteurs interviennent, il peut y avoir un ensemble plus large de situations concernant la force argumentative : par exemple un énoncé dont l'action implique un locuteur plus haut du point de vue hiérarchique ou plus actif en conversation peut être plus pertinent du point de vue argumentatif qu'un autre énoncé impliquant son émetteur. C'est pourquoi nous proposons, dans le paragraphe 8.3, une formalisation plus rigoureuse de la notion de force argumentative. Cette formalisation nous permettra par exemple d'inférer la pertinence de la génération du *mais* de relation directe (cf. paragraphe 8.2.2). Néanmoins, cet ordonnancement argumentatif nous permettra également d'inférer des connecteurs argumentatifs, tels que *parce que*, *car* ou *donc*.

8.3.1 Prémisses théoriques

Afin de formaliser de manière convenable l'idée d'ordonnancement argumentatif des énoncés (pour pouvoir calculer des connecteurs comme *mais* en emploi de relation directe – voir le paragraphe 8.2.2), nous nous appuyons sur le concept de *topos* dans la Théorie de l'argumentation de Anscombe et Ducrot [6], [58]. Dans un sens large, un *topos* est une règle argumentative partagée par une certaine communauté (qui n'a pas besoin d'avoir plus de membres que les interlocuteurs dans un dialogue, par exemple) [119]. Cette règle est utilisée afin d'autoriser le mouvement d'un énoncé (l'argument) vers une conclusion. Les *topoi* ont une nature scalaire, c'est-à-dire, ils ont une forme générale telle que 'Plus / moins l'objet *O* a la propriété *P*, plus / moins l'objet *O'* a la propriété *P'*'. Si nous notons par *P* la proposition 'L'objet *O* a la propriété *P*', par *Q* la proposition 'L'objet *O'* a la propriété *P'*', par + l'opérateur 'plus' et par – l'opérateur 'moins', alors il y a quatre *topoi* possibles (dénommées parfois des *formes topiques*, pour souligner leur caractère général) : (i) $\langle +P, +Q \rangle$, (ii) $\langle -P, -Q \rangle$, (iii) $\langle +P, -Q \rangle$, (iv) $\langle -P, +Q \rangle$. Si *P* est une proposition du type 'Le livre est intéressant' et *Q*, une proposition 'On lit le livre vite', les quatre formes topiques génèrent quatre arguments possibles : (Plus le livre est intéressant) $_{+P}$, '(plus on le lit vite) $_Q$ ', (ii) '(Moins le livre est intéressant) $_{-P}$, (moins on le lit vite) $_{-Q}$ ', (iii) '(Plus le livre est intéressant) $_{+P}$, (moins on le lit vite) $_{-Q}$ ', (iv) '(Moins le livre est intéressant) $_{-P}$, (plus on le lit vite) $_{+Q}$ '. De ces quatre *topoi*, (i) et (ii) sont compatibles l'un avec l'autre (c'est-à-dire, si on accepte (i) on accepte (ii) aussi, et inversement); pareillement, (iii) et (iv) sont des *topoi* compatibles, mais toute autre paire est constituée de *topoi* incompatibles. Ainsi, dans la TA, les formes topiques (i) et (ii) d'un côté, et (iii) et (iv) de l'autre côté, sont appelées *formes topiques équivalentes*. Les *topoi* (i) et (ii) sont appelés *formes topiques directes*, tandis que (iii) et (iv), *formes topiques inverses*.

Par la suite, Anscombe et Ducrot ont proposé une analyse plus fine des *topoi*, afin de répondre à certaines critiques, selon lesquelles leur théorie aurait abandonné complètement toute représentation vériconditionnelle des énoncés. Ainsi, d'une part, il y a les *topoi* qui sont déterminés automatiquement par les connaissances linguistiques et par les connaissances du sens

commun, par défaut, par exemple⁵ ‘Pierre est riche : il peut s’offrir tout ce qu’il veut’, où le fait que ‘on peut s’offrir tout ce qu’on veut’ est un corollaire *intrinsèque* de ‘on est riche’ est saisi par la forme topique $\langle +P, +Q \rangle$ (‘(Plus on est riche)_{+P}, (plus on peut s’offrir ce qu’on veut)_{+Q}’). De l’autre part, il y a des topoï qui doivent être appuyés par des connaissances *factuelles*, qui sont spécifiques au contexte où l’énoncé est produit, par exemple ‘Pierre est riche : il est donc avare’, où le fait que ‘on est avare’ n’est guère un corollaire de ‘on est riche’, mais peut être déterminé par d’autres topoï, *extrinsèques* au sens de ‘être riche’ et ‘être avare’ ; de tels topoï peuvent former une chaîne de la forme $\langle +P, -Q \rangle$ (‘(Plus on est riche)_P, (moins on donne)_{-Q}’), et $\langle -P, +Q \rangle$ (‘(moins on donne)_{-P}, (plus on est avare)_{+Q}’). Conséquemment, l’énoncé ‘Pierre est riche : il est donc avare’ peut être obtenu en *concaténant* deux topoï : le premier est plutôt factuel, parce qu’il ne peut être appliqué qu’à des entités *particulières* (par exemple, un certain ‘Pierre’) ; le deuxième est déterminé automatiquement par les sens de ‘avare’ et ‘donner’. Les topoï déterminés par des connaissances factuelles (comme, dans notre exemple, ‘Plus on est riche, moins on donne’) sont dénommés *extrinsèques* dans la terminologie de Anscombe [6] ; l’autre type de topoï, déterminé par des connaissances linguistiques (en fin de compte, *implicites*), sont dénommés *intrinsèques*.

8.3.2 Ordonnement argumentatif des topoï

Calcul des topoï

Des essais pour formaliser la Théorie de l’argumentation (TA) existent ; ils portent sur la correction et la complétude de la théorie [105], ou mettent en question ses fondements épistémiques [145]. Cependant, concernant l’utilisation *computationnelle* de la théorie, il y a apparemment peu de travaux : nous noterons le générateur ADVISOR II de Elhadad [60], où la notion de topoï argumentatifs est utilisée pour guider le choix lexical dans une application censée conseiller les étudiants dans leurs choix de sujets d’étude. Une autre utilisation computationnelle intéressante de la Théorie de l’argumentation, dans une approche qui la couple aussi avec la SDRT, réside dans l’utilisation de topoï (dans le sens de Ducrot [57]) pour calculer les SDRS [195]. Néanmoins, cette dernière application ne *formalise* pas à vrai dire des idées dans la Théorie de l’argumentation ; elle utilise juste cette dernière dans un cadre formel distinct.

Un premier problème à considérer lorsqu’on veut formaliser la TA de Anscombe et Ducrot concerne le calcul des topoï argumentatifs qui déterminent le lien entre deux propositions (une en tant qu’argument et l’autre, jouant le rôle de conclusion). Nous utilisons pour cela le même cadre logique du premier ordre que nous utilisons tout au long de cette thèse : les topoï sont considérés comme avoir les quatre formes suivantes : $\langle \alpha, \beta \rangle$ (forme topique directe positive), $\langle \neg\alpha, \neg\beta \rangle$ (forme topique directe négative), $\langle \alpha, \neg\beta \rangle$ (forme topique inverse négative), and $\langle \neg\alpha, \beta \rangle$ (forme topique inverse positive) ; ici, α et β désignent des *énoncés*. Dans cette représentation nous approximons les notions de la TA de PLUS (+) et MOINS (-) en considérant que + est réalisé en produisant l’énoncé affirmatif (noté par α , ou β) et que - est réalisé en produisant la forme *niée* de cet énoncé (donc, $\neg\alpha$ ou $\neg\beta$). Dans le même ordre d’idées, étant donnés deux énoncés α_1 et α_2 , nous devons décider s’ils peuvent faire partie (en tant qu’argument gauche) de topoï argumentatifs non-disjoints, c’est-à-dire, de topoï qui partagent la même conclusion (sous

⁵Les exemples dans ce paragraphe sont adaptés de [6].

forme positive ou négative). Plus précisément, deux topoï $\tau_1 := \langle \alpha_1, \beta_1 \rangle$ et $\tau_2 := \langle \alpha_2, \beta_2 \rangle$ sont dénommés *non-disjoints* si soit $\beta_1 = \beta_2$, soit $\beta_1 = \neg\beta_2$. Ces dernières conditions impliquent en fait que les valeurs de vérité de β_1 et β_2 satisfont, en tout modèle [63], les égalités précisées. Si deux topoï τ_1 et τ_2 sont non-disjoints, nous écrivons que $\tau_1 \cap \tau_2 = |\gamma|$, pour $\gamma = \beta_1$ et $\gamma = \beta_2$ ou $\gamma = \neg\beta_2$. Ainsi, α_1 et α_2 peuvent être comparés, d'un point de vue argumentatif, seulement s'ils servent d'arguments pour la même conclusion, ou pour des conclusions contraires. Cela revient au fait que les deux énoncés ne peuvent être comparés que s'il existe deux topoï non-disjoints qui ont ces deux énoncés comme leurs arguments gauches. Nous devons détecter ces topoï.

Pour ce faire, nous vérifions d'abord s'il existe un énoncé β dans la « conversation » dont α_1 et α_2 font partie, tel qu'il existe une relation rhétorique $\rho \in \{Consequence, Elabor_q\}$ qui a la propriété que soit $\rho(\alpha_1, \beta)$, soit $\rho(\alpha_2, \beta)$ est vraie (on note ce *desideratum* en utilisant une notation BNF (« Backus-Naur Form ») condensée – $\rho(\{\alpha_1|\alpha_2\}, \beta)$, où « | » spécifie les alternatives). Dans ce cadre, nous supposons que la structure rhétorique (une SDRS) du dialogue a été calculée au préalable (selon les démarches décrites dans les chapitres 3, 4 et 5) comme un graphe, où les nœuds sont les énoncés et les arêtes sont les relations rhétoriques entre ces énoncés. *Consequence* est une relation rhétorique monologique dans la SDRT ; cette relation relie deux énoncés dans le même tour de parole, tels que le deuxième énoncé est une conséquence vériconditionnelle du premier ; *Elab_q* est une relation rhétorique dialogique qui relie deux énoncés dans deux tours de parole différents (produits par des locuteurs distincts), tels que le deuxième énoncé est une question dont toute réponse élabore sur le premier énoncé (cette réponse est donc en relation monologique d'*Elaboration* avec le premier énoncé – cf. chapitre 3). L'utilisation de *Elab_q* est motivée par le fait que la question représentée par l'argument gauche de cette relation rhétorique peut être une conséquence argumentative du tour de parole antérieur : nous supposons deux locuteurs, *U* (le sujet humain) et *M* (la machine), qui sont engagés dans le dialogue suivant :

M : Vous pouvez emprunter encore trois livres, Monsieur !

U : OK, donc je peux prendre celui-ci aussi ?

M : Oui, vous pouvez l'emprunter, Monsieur.

Cette interaction contient une question de *U*, qui est dans une relation de *Elab_q* avec le premier énoncé de *M* ; la réponse de *M* est dans une relation d'*Elaboration* avec le premier énoncé, car, en effet, les deux tours de *M* accomplissent le même effet (concernant la tâche que le dialogue essaye d'aider à résoudre) qu'un tour unique de *M* :

M : Vous pouvez encore emprunter trois livres, donc vous pouvez prendre le livre 'X' ou celui que vous voulez.

où le livre 'X' et 'celui-ci' dans la question de l'utilisateur, réfèrent au même objet dans le monde physique. Si une telle relation est trouvée, alors nous supposons qu'un topos est trouvé : $\langle \{\alpha_1|\alpha_2\}, \beta \rangle$. Du fait que ce topos a été déterminé seulement à partir de la structure rhétorique, qui a été calculée à son tour en n'utilisant que l'ontologie de tâches (voir chapitres 3 et 4 dans cette thèse) et aucune information extrinsèque ou connaissance factuelle (c'est-à-dire, qui reflète l'état courant du dialogue) n'a été utilisée en plus des énoncés en tant que tels, il résulte que ce topos est *intrinsèque*. En notant par $\{\alpha_1|\alpha_2\}$ l'alternative qui n'a pas encore été explorée (si elle existe) en $\{\alpha_1|\alpha_2\}$, il résulte que nous devons vérifier si un des deux topoï $\langle \{\alpha_1|\alpha_2\}, \{\beta|\neg\beta\} \rangle$

existe. Cela peut se faire soit d'une manière semblable à ce qui a été décrit ci-dessus (c'est-à-dire, en cherchant une relation rhétorique appropriée), soit à partir des bases de connaissances disponibles.

En ce qui concerne les bases de connaissances qui sont utilisées dans notre travail, elles intègrent trois types d'informations :

- (i) connaissances générales concernant le domaine de l'application (par exemple, la réservation de livres dans une bibliothèque) ;
- (ii) des connaissances factuelles concernant le statut particulier de certains éléments d'une tâche particulière (par exemple, la situation du compte courant d'un client d'une bibliothèque – c'est-à-dire, le nombre de livres déjà empruntés par ce client, etc.) ;
- (iii) des connaissances factuelles à l'égard du dialogue courant (c'est-à-dire, l'historique du dialogue courant).

Les connaissances du type (i) sont structurées sous la forme d'une ontologie de tâches [35] où des faits généraux avec les relations entre eux sont exprimés (par exemple, 'On peut emprunter cinq livres à la fois', 'Plus de livres on a emprunté, moins de livres on peut encore emprunter', etc.). Les connaissances du type (ii) sont structurées d'une manière semblable à l'ontologie de tâches (on appelle donc ces connaissances « ontologie de l'utilisateur » ; une telle ontologie contient, pour un utilisateur donné, des informations comme 'Le client X a emprunté trois livres, deux DVDs et trois CDs audio'). Les connaissances du type (iii) contiennent les sémantiques des énoncés auxquels les partenaires de dialogue se sont engagés de manière explicite ; ces connaissances sont structurées comme un ensemble de *listes d'engagements* : pour chaque utilisateur U_i dans un dialogue où N locuteurs prennent part, il existe une liste d'engagements CS_i (CS – « Commitment Store » en anglais) qui contient les sémantiques des énoncés que U_i a produits, avec les sémantiques des énoncés produits par les autres partenaires dans le dialogue et auxquels U_i s'est également engagé (c'est-à-dire, il ne les a pas infirmés de manière explicite ou implicite ; cela est indiqué par les relations rhétoriques entre ces énoncés et les énoncés de U_i – voir chapitre 7 pour des détails sur les listes d'engagements), et enfin, les énoncés des autres interlocuteurs, avec lesquels U_i n'est explicitement pas d'accord, avec les relations rhétoriques qui signalent ce fait (par exemple, *P-Corr* ou *Contrast* – voir chapitres 3 et 7).

La manière dont les listes d'engagements sont définies est inspirée des travaux de Muller, Prévot et Maudet [112] ; les détails sur ce point sont précisés dans le chapitre 7 ; pour l'exemple ci-dessus, la liste d'engagements de U , après avoir posé la question, est l'ensemble :

$$CS_U = \{K(\pi_1), K(\pi_2), \Sigma_{Elab_q(\pi_1, \pi_2)}\},$$

où π_1 et π_2 désignent le premier énoncé de M et le premier énoncé de U (la question) respectivement, la fonction $K(\pi)$ retourne la sémantique (c'est-à-dire, l'acte locutoire) de l'énoncé π (voir chapitres 3 et 7 sur cette notion), et $\Sigma_{\rho(\pi_1, \pi_2)}$ désigne la sémantique de la relation rhétorique $\rho(\pi_1, \pi_2)$; comme nous l'avons vu dans les chapitres 3 et 4 de cette thèse, cette sémantique est une règle à la Prolog qui spécifie un ensemble de conditions qui doivent être satisfaites par les énoncés pour qu'ils soient reliés par cette relation rhétorique. Pour une présentation détaillée de ces sémantiques, ainsi que du cadre qui est à la base de ces dernières, nous renvoyons le lecteur au chapitre 3. Cependant, nous ajoutons ici un détail important sur la *persistance* des connaissances : tandis que les connaissances du type (i) et (ii) sont persistantes⁶, les connais-

⁶Il faut toutefois noter que les connaissances du type (ii) reliées à un utilisateur particulier sont persistantes

sances du type (iii) sont volatiles au sens où lorsque chaque nouveau dialogue commence, les listes d'engagements de tous les partenaires dans le dialogue sont initialisées avec l'ensemble vide⁷.

Ces bases de connaissances jouent un rôle essentiel non seulement parce qu'elles nous permettent de calculer les topoï argumentatifs qui ne sont pas résultés à partir de la structure rhétorique, mais aussi parce qu'elles nous aident à distinguer entre les topoï intrinsèques et extrinsèques, d'une manière moins « linguistique » par rapport aux tests de Anscombe : nous posons qu'un topos qui est soit résulté à partir d'une structure rhétorique (comme nous l'avons montré ci-dessus), soit calculé en n'utilisant que les connaissances du type (i) ci-dessus (c'est-à-dire, des connaissances générales concernant le domaine de l'application) est intrinsèque. Inversement, un topos calculé en utilisant des connaissances des types (ii) et (iii) ci-dessus (c'est-à-dire, concernant l'utilisateur (s'il est déjà enregistré comme client dans la bibliothèque), et le dialogue courant) est extrinsèque.

De plus, nous stipulons aussi que les topoï intrinsèques peuvent être classés en trois catégories, selon leur nature :

- des topoï *conventionnels* (ou « doux ») : ceux-ci sont déterminés par des conventions *ad-hoc*, telles qu'imposer une certaine limite supérieure au nombre de livres qu'un client peut réserver dans une bibliothèque ; un exemple d'un tel topos est :

(plus le nombre des livres déjà réservés par un client est près de cinq)₊, (moins ce client peut encore réserver de livres)₋.

- des topoï *déontiques* : ceux-ci sont toujours déterminés par des conventions, mais d'un type particulier, comme des engagements ou des promesses vers un autre interlocuteur ; un tel topos est :

(plus un livre a déjà été réservé par un client)₊, (moins on peut prêter ce livre à un autre client)₋.

- des lois naturelles (ou topoï « durs ») : celles-ci correspondent à des faits ou des contraintes naturels ; ils tendent à être surtout de la forme $\langle \pm P, -Q \rangle$; un exemple d'un tel topos est :

(plus un livre est absent de la bibliothèque)₊, (moins un peut prêter ce livre à un client)₋.

Les topoï intrinsèques conventionnels et déontiques, tout comme les topoï extrinsèques, sont plutôt graduels dans leur nature (c'est-à-dire, du type 'plus / moins *P*, plus / moins *Q*') ; cette gradualité revient au fait qu'une préférence extrinsèques (subjective) d'un locuteur peut être suivie de manière plus ou moins stricte, tout comme une promesse ou un engagement, qui peuvent être tenus de façon plus ou moins stricte. De l'autre part, les topoï intrinsèques « durs » sont plutôt strictes en ce qu'ils spécifient ce qu'il *ne* peut *pas* être fait, par exemple, si un certain livre n'est pas présent dans la bibliothèque, alors il ne peut *pas* être prêté à un client, tout simplement, il n'y a pas de place pour la gradualité.

seulement autant que cet utilisateur est un client de la bibliothèque.

⁷Cela est une différence essentielle par rapport au dialogue *multi-session*, où la machine garde les connaissances du type (iii) disponibles d'une session de dialogue à l'autre, et le dialogue *multi-locuteurs*, concerné par les traitements décrits dans cette thèse.

Ces idées nous amènent à observer le fait que, pour deux topoï disjoints $\tau_1 := \langle \alpha_1, \beta_1 \rangle$ et $\tau_2 = \langle \alpha_2, \beta_2 \rangle$ tels qu'il existe un γ (explicite ou implicite – voir ci-dessous) ainsi que $\tau_1 \cap \tau_2 = |\gamma|$, les topoï induisent l'ordonnement argumentatif suivant sur les énoncés :

- Si τ_1 est un topos intrinsèque conventionnel et τ_2 est un topos extrinsèque, alors $\alpha_1 < \alpha_2$, où $<$ désigne la relation d'*ordre argumentatif*, c'est-à-dire, α_1 a une force argumentative moindre par rapport à α_2 pour la conclusion $\{\gamma | \neg\gamma\}$. Autrement dit, si deux arguments déterminent une conclusion *via* un topos extrinsèque et, respectivement, *via* un topos intrinsèque conventionnel, alors l'argument qui détermine la conclusion (ou sa négation) *via* le topos extrinsèque est plus fort que l'argument qui détermine cette conclusion *via* le topos intrinsèque conventionnel. Cela est justifié par le fait qu'un argument qui est appuyé par des connaissances factuelles concernant la situation de dialogue courante est plus fort qu'un argument concernant des vérités générales, *conventionnelles*. Par exemple, nous supposons la même tâche de réservation de livres, où la machine dit à un utilisateur qui a déjà emprunté cinq livres (c'est-à-dire, le nombre maximal permis) que, étant donnée sa très bonne conduite (il n'a jamais eu de retards pour le retour des documents, etc.), elle va lui donner un sixième livre :

M : Vous avez déjà emprunté cinq livres, Monsieur, *mais* cette fois-ci je ferai une exception et je vous donnerai aussi le livre 'X'.

Ici, le topos intrinsèque conventionnel 'plus on a déjà emprunté cinq livres, moins un peut encore emprunter des livres' détermine la conclusion 'Vous (le client) ne pourrez emprunter le livre 'X' que si vous rendez au moins un des livres qu'on vous a déjà prêtés'. De l'autre côté, des informations factuelles 'Cette personne a été un bon client de la bibliothèque', avec le topos extrinsèque (extrinsèque, parce qu'il n'est dicté par aucune règle par défaut, mais par les préférences subjectives du bibliothécaire) 'Si un client est bon, alors on pourra faire des exceptions et lui prêter parfois plus de cinq livres' (ce qui est une instance de la forme topique extrinsèque 'Plus un client est bon, plus les chances qu'on le laisse emprunter plus de cinq livres sont élevées') déterminent la conclusion 'Cette fois-ci je laisserai le client (« vous » – le destinataire de l'énoncé de *M*) emprunter aussi le livre 'X', même s'il a déjà emprunté cinq livres'.

- Si τ_1 est un topos déontique et τ_2 est un topos extrinsèque, alors $\alpha_2 < \alpha_1$, c'est-à-dire, l'énoncé qui détermine une conclusion *via* un topos intrinsèque déontique est argumentativement plus fort qu'un énoncé qui détermine la même conclusion (ou sa forme niée) *via* un topos extrinsèque. Cela est justifié par le fait que, normalement, une promesse ou un engagement vers quelqu'un prévaut sur les préférences subjectives du moment. Autrement dit, les engagements (ou les promesses) doivent être tenu(e)s avant que les buts subjectifs puissent être satisfaits ; cela semble une « règle » pertinente pour la machine : le dialogue suivant entre la machine *M* et un client *U* illustre ce point de vue :

*U*₁ : Bonjour, je voudrais réserver le livre 'X'.

*M*₁ : Bonjour, OK, un instant... Eh ben, en fait vous avez déjà emprunté cinq livres...

*U*₂ : Oui, *mais* j'ai un examen demain ! Pourriez-vous m'aider s'il vous plaît ?

*M*₃ : Eh ben, cette fois-ci je pourrais faire une exception, *parce que* vous n'avez

jamais été en retard avec la restitution des livres... *mais* le livre 'X' a déjà été réservé par un autre client...

Dans cet exemple, dans son premier tour de parole, la machine réfute le plan de U en s'appuyant sur le topos intrinsèque conventionnel 'Plus on a déjà emprunté cinq livres, moins on peut encore emprunter des livres'. Ensuite, dans son deuxième tour de parole, l'utilisateur soutient son plan *via* le topos extrinsèque 'Plus on a un examen, plus on a besoin d'un certain livre'; ce topos est extrinsèque parce qu'il est paramétré en même temps par un livre particulier ('un *certain* livre') et par la corrélation particulière entre l'examen et le besoin de U pour le livre; cela explique ainsi l'utilisation de *mais* en U_2 . Dans son troisième tour de parole, la machine concède qu'elle peut faire une exception, parce que U a été un bon client; cela est acquis toujours *via* un topos extrinsèque 'Plus on est un bon client, plus il est probable qu'on puisse emprunter plus de livres que le nombre maximal *conventionnel* de cinq livres'. Cependant, cela est argumentativement plus faible que le fait que le livre a déjà été réservé par un autre client, ce qui justifie l'utilisation de *mais* dans M_3 . Cette relation argumentative est justifiée par le fait que le topos 'Plus un certain livre est réservé par un client, moins on peut l'emprunter à un autre client' est déontique, donc plus fort que le topos extrinsèque utilisé précédemment.

- Si τ_1 est un topos intrinsèque déontique et τ_2 est une loi naturelle (c'est-à-dire, un topos dur), alors $\alpha_1 < \alpha_2$, c'est-à-dire, l'énoncé qui détermine une conclusion *via* une loi naturelle est argumentativement plus fort qu'un énoncé qui détermine la même conclusion *via* un topos déontique. Cela est justifié par le fait que les contraintes imposées par la nature sont plus fortes que les engagements personnels (ou sociaux). Autrement dit, un fait dans la nature (qui apparaît, par exemple, après qu'un engagement a été fait) peut déterminer l'incapacité de celui qui s'était engagé, de tenir ses promesses. Par exemple, dans le dialogue suivant un utilisateur a réservé un livre, mais, de toutes façons, il ne peut pas prendre ce livre parce que l'utilisateur qui l'avait précédemment emprunté ne l'a pas encore rendu à la bibliothèque :

U_1 : Bonjour, j'ai réservé le livre 'X' il y a une semaine, et vous m'avez dit de venir le récupérer aujourd'hui...

M_1 : Oui, effectivement, *mais* le client qui avait emprunté le livre avant ne l'a pas encore rendu, *donc* je ne peux pas vous le donner aujourd'hui...

Dans cet exemple, l'utilisateur demande d'abord le livre 'X', en utilisant le topos intrinsèque déontique 'Plus on a réservé un livre, plus il est probable qu'on puisse le récupérer à la date fixée de commun accord avec le bibliothécaire'; néanmoins, la machine réfute cet argument *via* le topos dur plus puissant 'Plus un client ne rend pas un livre à l'heure, moins on est capable de le prêter à d'autres clients', où 'plus' peut en fait être remplacé par 'si', et 'moins', par 'ne... pas'; cela justifie l'utilisation de *mais* dans le tour de parole de M . De plus, la machine souligne plus loin ce fait, en affirmant la conclusion de son argument; cette conclusion représente la forme niée de la conclusion de l'argument de U , ce qui justifie l'utilisation de *donc* dans le tour de M . Dans ce tour, la machine rend en fait *explicite* tout le topos dur qu'elle utilise.

Il est vrai que dans cet exemple, le connecteur *mais* (de relation indirecte) peut être calculé

sans utiliser les topoï ; il en va de même pour *donc*, qui résulte d'une relation rhétorique de *Consequence* entre les énoncés reliés par ce connecteur. Cependant, les topoï offrent un cadre plus général pour traiter ce type d'exemples, ainsi que les autres situations, illustrées le long de ce paragraphe.

En vertu de cette discussion, nous pouvons conclure que les topoï peuvent être, eux aussi, ordonnés du point de vue argumentatif, selon leurs types, dans la progression suivante (du plus faible au plus puissant) : (i) topoï intrinsèques conventionnels, (ii) topoï extrinsèques, (iii) topoï intrinsèques déontiques, (iv) topoï intrinsèques durs. En notant par τ_I^{soft} les topoï du type (i), par τ_E les topoï du type (ii), par τ_I^{deontic} les topoï du type (iii), et par τ_I^{hard} , les topoï du type (iv), nous pouvons écrire, de manière résumée :

$$\tau_I^{\text{soft}} < \tau_E < \tau_I^{\text{deontic}} < \tau_I^{\text{hard}}.$$

De plus, nous stipulons que, à l'exception des topoï intrinsèques déontiques, tous les autres topoï ne sont pas scalaires, c'est-à-dire, deux topoï du même type ont la même force argumentative. Cela est justifié par le fait que, par exemple, les conventions (sociales) ne sont pas intrinsèquement structurées de manière hiérarchique, ou que les préférences (subjectives) extrinsèques ne sont pas non plus structurées de manière hiérarchique, d'un point de vue extérieur (c'est-à-dire, seul le locuteur concerné donne des priorités diverses à ses préférences subjectives), et, enfin, les lois naturelles sont « égales dans leur force » (par exemple, l'absence d'un livre est aussi forte que l'absence du bibliothécaire, comme un argument pour l'impossibilité d'emprunter ce livre-là).

Cependant, en ce qui concerne les topoï intrinsèques déontiques, les choses sont un peu différentes, car les hiérarchies sociales peuvent structurer ces topoï de manière hiérarchique aussi. Par exemple, le dialogue suivant peut en fin de compte amener un bibliothécaire à donner à un client *U* un livre déjà réservé par un autre client :

*U*₁ : Bonjour, je voudrais le livre 'X' s'il vous plaît !

*M*₁ : Je suis désolé, ce livre a déjà été réservé par un étudiant...

*U*₂ : OK, *mais* moi je suis Professeur des Universités ! Pourriez-vous *donc* faire quelque chose et me prêter ce livre pour mon cours ?

*M*₂ : Eh ben, je vais annuler sa réservation !

Dans son premier tour de parole, la machine réfute le plan de *U*, en s'appuyant sur le topos intrinsèque déontique 'Plus on s'est engagé à emprunter le livre à un client, moins on peut donner ce livre-là à un autre client'⁸. Cependant, dans son deuxième tour de parole, *U* défend son plan initial, *via* un autre topos intrinsèque déontique, 'Plus on est situé haut dans la société, plus il est probable qu'on soit servi, par rapport à d'autres personnes, situées moins haut dans la hiérarchie' ; cela justifie l'utilisation de *mais* dans *U*₂, parce que le deuxième topos déontique est argumentativement plus fort que le premier topos déontique. Dans son deuxième tour de parole, la machine confirme ce topos, en tirant la conclusion que 'le livre peut être prêté à *U*'.

L'ordonnement entre les topoï intrinsèques déontiques est néanmoins déterminé en utilisant des informations factuelles (extrinsèques), concernant les profils des utilisateurs (ces profils contiennent aussi les statuts sociaux de ces utilisateurs).

⁸Dans le contexte de l'application « bibliothèque virtuelle », un 'engagement à emprunter un livre à un client' est équivalent au fait que 'ce client a réservé ce livre'.

Afin de voir comment les topoï extrinsèques ou intrinsèques (non déterminés par la structure rhétorique) peuvent être repérés, nous avons besoin de présenter à un certain niveau de détail la manière dont les énoncés sont codés. Essentiellement, nous utilisons deux niveaux de codage : un niveau sémantique (logique), pour représenter les intentions communicationnelles provenant du contrôleur de dialogue (voir chapitres 1, 3, 4, 5, 6, mais aussi [35]), et un niveau syntaxique, pour manipuler les énoncés lorsque nous utilisons les formes logiques de ceux-ci en génération de surface.

Si, par exemple, l'énoncé (étiqueté π) 'Il y a eu de lents changements en Astronomie dans les deux derniers siècles' doit être produit par M , le contrôleur de dialogue aura fourni (au générateur) une forme logique comme :

$$\exists X, Y, Z, T : \text{object}(X) \wedge \text{modif}(X) \wedge \text{feature}(\text{modif}, Y) \wedge \text{equals}(X, \text{'subject : :astronomy'}) \wedge \text{equals}(Y, \text{'small'}) \wedge \text{equals}(\Delta t(\pi), Z) \wedge \text{equals}(Z, \text{'year'}) \wedge \text{value}(Z, T) \wedge \text{equals}(T, \text{'-200'}) \wedge \text{equals}(t(\pi), t).$$

Ici, le prédicat $\text{object}/1$ est vrai si son argument est l'objet de l'énoncé π ; $\text{modif}/1$ désigne le prédicat de l'énoncé π , c'est à dire, le fait de modifier des caractéristiques de l'entité prise comme argument; $\text{feature}/2$ est vrai si son deuxième argument est une caractéristique de son premier argument; $\text{equals}/2$ est un prédicat discursif qui a été défini dans le chapitre 3, mais pour la complétude de l'exposition nous rappelons que ce prédicat est vrai si ses deux arguments sont identiques (c'est-à-dire, ils ont la même valeur); $\Delta t/1$ est une fonction qui retourne le temps (moment ou intervalle) *concerné par* l'énoncé (c'est-à-dire, dans notre cas, deux siècles); $t/1$ est une fonction qui retourne le temps de l'énoncé, c'est-à-dire, le moment où l'énoncé est produit (d'habitude, ce temps est le moment présent – cela est noté par t , mais il peut aussi être passé ou futur, pour des énoncés comme 'Hier, j'ai entendu Pierre dire : Il y a eu des changements en Astronomie dans les deux derniers siècles' – dans ce cas nous avons deux formules logiques imbriquées (l'objet de la première est représenté par la dernière, dont le temps de production est le passé, noté par $t_.$); enfin, $\text{value}/2$ attribue une valeur numérique (représentée par le deuxième argument) à l'entité représentée dans le premier argument.

A partir de cette description, il est plutôt évident qu'il n'y a pas de correspondance un-à-un entre la forme logique et la réalisation linguistique; c'est précisément pour cette raison que nous contrôlons le passage de la première vers la dernière *via* un niveau intermédiaire, que nous appelons « syntaxique » : essentiellement, chaque énoncé est représenté comme un ensemble ordonné de paires attribut-valeur de la forme :

$$([\text{agt}] = v_1) \uparrow (\langle \text{pred} \rangle = v_2) \uparrow ([\text{mod}]_{\langle p \rangle} = m_i) * \uparrow ([\text{obj}] = v_3) \uparrow ([\text{mod}]_{[\text{o}]} = m_j) * \uparrow ([\text{pat}] = v_4).$$

Dans cette notation, \uparrow désigne l'opérateur de concaténation, $[\text{agt}]$, $\langle \text{pred} \rangle$, $[\text{obj}]$ et $[\text{pat}]$ désignent respectivement l'agent, le prédicat, l'objet et le patient (c'est-à-dire, l'entité qui subit les effets d'une action). $[\text{mod}]_{\langle p \rangle}$ et $[\text{mod}]_{[\text{o}]}$ désignent les modificateurs du prédicat et, respectivement, de l'objet; le symbole $*$ indique le fait qu'il peut y avoir zéro, un ou plusieurs modificateurs (du type spécifié à gauche de l'astérisque). v_1 à v_4 et les ensembles m_i et m_j désignent les valeurs prises par les attributs à gauche. Dans notre exemple, la transcription « syntaxique » devient :

$$([\text{agt}] = \emptyset) \uparrow (\langle \text{pred} \rangle = \text{'modif'}) \uparrow ([\text{mod}]_{\langle p \rangle} = \text{'small'}) \uparrow ([\text{mod}]_{\langle p \rangle} = \text{'timespan'}) \uparrow ([\text{mod}]_{[\text{mod}]_{\langle p \rangle}^{(2)}} = \text{'years'}) \uparrow ([\text{mod}]_{[\text{mod}]_{[\text{mod}]_{\langle p \rangle}^{(2)}}} = \text{'-200'}) \uparrow ([\text{obj}] = \text{'subject : :astronomy'}) \uparrow$$

([pat] = \emptyset).

Dans cette représentation, \emptyset indique le fait que l'attribut correspondant n'est pas initialisé, $[\text{mod}]_{(p)}^{(2)}$ désigne le deuxième modificateur du prédicat, tandis que $[\text{mod}]_{\mu}$ désigne le modificateur du modificateur μ ; cela explique la prolifération des indices pour les modificateurs.

Pour déterminer, pour un énoncé π , ses topoï argumentatifs qui ne résultent pas de la structure rhétorique, nous nous appuyons sur trois sources de connaissances (l'ontologie de tâches, l'ontologie factuelle des utilisateurs, et les listes d'engagements des partenaires du dialogue). Afin de déterminer les topoï intrinsèques, nous réalisons une suite de lambda-abstractions sur tous les attributs dans la représentation « syntaxique » des énoncés, à l'exception de l'objet ([obj]) et du prédicat ($\langle \text{pred} \rangle$) ; cette représentation « syntaxique » devient ainsi :

$$\lambda. [\text{agt}] \uparrow (\langle \text{pred} \rangle = v_2) \uparrow \lambda. [\text{mod}]_{(p)}^* \uparrow ([\text{obj}] = v_3) \uparrow \lambda. [\text{mod}]_{[o]}^* \uparrow \lambda. [\text{pat}].$$

Nous notons une telle abstraction par $\tilde{\pi}$. Cela est nécessaire afin que nous puissions appairer les énoncés en entrée aux faits généraux dans l'ontologie de tâches (c'est-à-dire, aux topoï intrinsèques de la forme 'Plus / moins on fait quelque chose, plus / moins on obtient quelque chose'). Dans notre énoncé pris comme exemple, les lambda-abstractions déterminent⁹ :

$$([\text{agt}] = \emptyset) \uparrow (\langle \text{pred} \rangle = \text{'modif'}) \uparrow \lambda. [\text{mod}]_{(p)} \uparrow \lambda. [\text{mod}]_{(p)} \uparrow \lambda. [\text{mod}]_{[\text{mod}]_{(p)}^{(2)}} \uparrow \lambda. [\text{mod}]_{[\text{mod}]_{[\text{mod}]_{(p)}^{(2)}}} \uparrow ([\text{obj}] = \text{'subject : :astronomy'}) \uparrow ([\text{pat}] = \emptyset).$$

Au niveau linguistique, une telle forme peut être appariée à un topos intrinsèque de la forme 'Moins les changements dans un domaine sont rapides, plus il est probable que des vieux livres sur ce sujet restent utiles', ou un topos extrinsèque de la forme 'Plus les changements dans un domaine sont rapides, plus il est probable qu'on soit intéressé à ce domaine' qui est basé, par exemple, sur la liste d'engagements d'un certain locuteur U dans un dialogue courant, où cet usager aurait pu dire 'Ô, j'adore les choses nouvelles, qui changent rapidement !' (les indices inférieurs désignent des tours de parole distincts) :

U_1 : Bonjour, je voudrais un livre sur quelque chose de nouveau, comme l'Astronomie ! J'adore les choses nouvelles, qui changent rapidement !

M_1 : Bonjour, Monsieur ! OK, que désirez vous, plus précisément ? Une introduction générale au domaine, ou un livre plus spécialisé ?

U_2 : Eh ben, je ne sais pas... Je commencerais avec un bouquin plutôt basique, plutôt ancien ; l'Astronomie, de toutes façons, n'a pas trop changé dans les deux dernières siècles...

M_2 : En fait, il y a eu de lents changements en Astronomie pendant les deux dernières centaines d'années, Monsieur. Voyons voir, je vous propose un livre sur l'As-trophysique ; là, les progrès récents sont très importants.

En ce qui concerne les différents types de topoï intrinsèques, ils peuvent être évalués en termes des connaissances utilisées pour les déterminer. Ainsi, dans l'ontologie de tâches, les règles qui précisent les relations entre les entités sont de deux types :

- (i) des règles *naturelles* (c'est-à-dire, du type 'Si un objet n'est pas disponible, alors on ne peut pas le donner à quelqu'un'), et

⁹On devrait noter que les attributs non-initialisés (c'est-à-dire, les attributs qui ont la valeur \emptyset) ne subissent pas de lambda-abstraction.

(ii) des règles *conventionnelles* (c'est-à-dire, du type 'Si une personne a sa carte d'abonnement, alors on peut octroyer à cette personne l'accès à un certain service').

Conséquemment, si un topos a été déterminé en n'utilisant que des règles naturelles, alors il est un topos intrinsèque dur ; sinon, s'il a été déterminé en utilisant aussi des règles conventionnelles, il est un topos intrinsèque conventionnel.

Quant aux topos déontiques, ils ne peuvent être déterminés qu'à partir de l'ontologie des utilisateurs : si dans le profil d'un certain client nous trouvons l'information que la machine s'était engagée à réserver un certain livre 'X' pour ce client, alors ce livre ne peut pas être prêté à un autre client. D'un point de vue procédural, pour chaque nouvel objet en discussion (par exemple, un livre), la machine étudie si l'objet a un marqueur précisant que cet objet a déjà été sujet d'un engagement vers un client ; si cela est le cas, alors nous autorisons le topos déontique approprié, concernant cet objet et ce client.

Algorithme d'ordonnement

Dans le paragraphe antérieur nous avons présenté de manière détaillée tous les « ingrédients » dont nous avons besoin pour construire une procédure pour évaluer automatiquement les énoncés, en termes de leur force argumentative. Ainsi, une telle procédure peut être précisée comme ci-dessous :

pour deux énoncés libellés α_1 et α_2 :

1. **s'il** existe un énoncé β dans le dialogue courant, tel que'il existe une relation rhétorique ρ dans la SDRS du dialogue, avec $\rho \in \{Consequence, Elab_q\}$, telle que $\rho(\{\alpha_1|\alpha_2\}, \beta)$ est vraie, **alors** :

(a) choisir le topos $\tau_{\{1|2\}} = \langle \{\alpha_1|\alpha_2\}, \beta \rangle$:

- i. le topos $\tau_{\{1|2\}}$ est intrinsèque (voir le paragraphe antérieur) ;
- ii. **si** seules des connaissances naturelles dans l'ontologie de tâches sont utilisées, alors ce topos sera intrinsèque dur ;
- iii. **sinon, si** nous avons utilisé aussi des connaissances conventionnelles dans l'ontologie de tâches, éventuellement avec des connaissances concernant l'utilisateur (dans l'ontologie des utilisateurs), **et** les objets dans ce topos n'ont pas de marqueurs déontiques, **alors** ce topos est intrinsèque conventionnel ;
- iv. **sinon** (c'est-à-dire, si les objets concernés par le topos ont des marqueurs déontiques), alors le topos est intrinsèque déontique ;

(b) chercher un topos du type $\tau_{\overline{\{1|2\}}} = \langle \overline{\{\alpha_1|\alpha_2\}}, \{\beta|\neg\beta\} \rangle$:

- i. **si** ce topos est déterminé en n'utilisant que des connaissances générales dans l'ontologie de tâches, ou des connaissances persistantes concernant l'utilisateur qui a produit $\overline{\{\alpha_1|\alpha_2\}}$, **alors** :
 - A. le topos $\tau_{\overline{\{1|2\}}}$ est intrinsèque (voir le paragraphe antérieur) ;
 - B. **si** seules des connaissances du type naturel dans l'ontologie de tâches sont utilisées, alors ce topos sera intrinsèque dur ;

- C. **sinon, si** nous utilisons aussi des connaissances du type conventionnel dans l'ontologie de tâches, éventuellement avec des connaissances concernant l'utilisateur, et les objets dans ce topos n'ont pas de marqueurs déontiques, alors ce topos est intrinsèque conventionnel ;
- D. **sinon** (c'est-à-dire, si les objets concernés par le topos ont des marqueurs déontiques), alors le topos est intrinsèque déontique ;
- ii. **sinon, si** nous avons besoin d'utiliser des informations dans les listes d'engagements des partenaires du dialogue, alors le topos $(\tau_{\overline{\{1|2\}}})$ est extrinsèque ;
- iii. **sinon** (c'est-à-dire, le topos $\tau_{\overline{\{1|2\}}}$ ne peut pas être déterminé), **alors aller à** l'étape 2 ;
2. **sinon**, chercher deux topoï non-disjoints pour α_1 et α_2 , qui ont comme conclusions un énoncé (ou sa forme niée) soit dans l'ontologie de tâches, soit dans les listes d'engagements des locuteurs :
- (a) réaliser des lambda-abstractions sur les agents et les patients de α_1 et α_2 , selon la discussion dans le paragraphe antérieur, pour obtenir les énoncés « abstraits » $\tilde{\alpha}_1$ et $\tilde{\alpha}_2$;
- (b) **aller à** l'étape 1.(b), en prenant $\tilde{\alpha}_1$ et $\tilde{\alpha}_2$ en entrée ; donc, déterminer deux topoï non-disjoints pour ces énoncés ;
3. **si** le topos $\tau_{\{1|2\}} = \langle \{\tilde{\alpha}_1|\tilde{\alpha}_2\}, \{\beta|\neg\beta\} \rangle$ est intrinsèque et le topos $\tau_{\overline{\{1|2\}}} = \langle \{\tilde{\alpha}_1|\tilde{\alpha}_2\}, \{\beta|\neg\beta\} \rangle$ est extrinsèque, **alors** :
- (a) **si** $\tau_{\{1|2\}}$ est intrinsèque conventionnel, **alors** $\{\alpha_1|\alpha_2\} < \overline{\{\alpha_1|\alpha_2\}}$; **STOP** ;
- (b) **sinon** (c'est-à-dire, $\tau_{\{1|2\}}$ est intrinsèque déontique ou dur), **alors** $\overline{\{\alpha_1|\alpha_2\}} < \{\alpha_1|\alpha_2\}$; **STOP** ;
4. **sinon, si** les deux topoï $\tau_{\{1|2\}}$ et $\tau_{\overline{\{1|2\}}}$ sont soit intrinsèques ou extrinsèques :
- (a) **si** les deux topoï sont intrinsèques déontiques :
- i. **si** l'agent *humain* (c'est-à-dire, un utilisateur / client dans l'application de « bibliothécaire virtuel ») dans la prémisses et / ou la conclusion du topos $\tau_{\{1|2\}}$ à une position sociale moins haute que l'agent humain dans la prémisses et / ou conclusion du topos $\tau_{\overline{\{1|2\}}}$, **alors** $\{\alpha_1|\alpha_2\} < \overline{\{\alpha_1|\alpha_2\}}$; **STOP** ;
- ii. **sinon** (c'est-à-dire, les agents humains dans la prémisses et / ou la conclusion des deux topoï ont la même position sociale, **alors** $\overline{\{\alpha_1|\alpha_2\}} \equiv \{\alpha_1|\alpha_2\}$; **STOP** ;
- (b) **sinon** (c'est-à-dire, les deux topoï sont soit extrinsèques, soit intrinsèques conventionnels ou durs) **alors** $\overline{\{\alpha_1|\alpha_2\}} \equiv \{\alpha_1|\alpha_2\}$; **STOP** ;
5. **sinon** (c'est-à-dire, il n'y a pas de topoï non-disjoints pour les énoncés α_1 et α_2) **alors** les deux énoncés ne peuvent pas être comparés du point de vue argumentatif : $\{\alpha_1|\alpha_2\}!\overline{\{\alpha_1|\alpha_2\}}$ (le point d'exclamation marque l'absence d'une relation d'ordre argumentatif entre les deux énoncés) ; **STOP**.

La présentation de cette procédure aurait été plus lisible mais, à notre avis, moins rigoureuse, si nous avions écrit α_1 au lieu de $\{\alpha_1|\alpha_2\}$ et α_2 au lieu de $\overline{\{\alpha_1|\alpha_2\}}$. Cet algorithme synthétise, d'une façon concise, toute la discussion dans le paragraphe antérieur sur le calcul des topoï, mais offre un moyen de tracer facilement le mécanisme d'ordonnement argumentatif des énoncés.

8.3.3 Exemple en dialogue multi-locuteurs

L'algorithme présenté dans le paragraphe précédent peut être utilisé à plusieurs niveaux pour contrôler la génération des réponses dans des applications de dialogue à plusieurs interlocuteurs. Afin d'illustrer une partie de ces utilisations, nous considérons toujours notre application de dialogue dans le domaine de la réservation de livres. Ainsi, nous considérons ci-dessous un exemple de dialogue entre une machine M et deux utilisateurs $U^{(1)}$ et $U^{(2)}$ qui veulent tous les deux emprunter un livre, mais ils sont quand même traités différemment par M . Le dialogue est analysé ci-dessous (les indices inférieurs désignent les tours de parole successifs) :

$U_1^{(1)}$: Je voudrais emprunter une tragédie ancienne.

$U_1^{(2)}$: Moi aussi, de Sophocle.

M_1 (en parlant aux deux clients) : Je ne peux plus rien vous prêter, *parce que* vous avez déjà emprunté cinq livres...

$U_2^{(1)}$: Oui, *mais* nous avons un examen demain, pourriez-vous faire une exception ?

M_2 (en parlant à $U^{(1)}$) : Pour vous, oui, *car* vous êtes un bon client

M_2 (en parlant à $U^{(2)}$) : *mais pas* pour vous, *parce que* je n'ai plus qu'un exemplaire dans la bibliothèque, et que vous n'avez pas encore rendu les livres que vous avez empruntés il y a cinq mois !

$U_2^{(2)}$: Pourrais-je *quand même* lire le livre dans la bibliothèque, en consultation sur place ?

Dans cet exemple nous pouvons observer comment la machine parle à deux clients, en les traitant différemment, selon leurs « histoires » propres : au début, la machine utilise le topos intrinsèque conventionnel 'Plus on a emprunté plus de cinq livres, moins on peut encore emprunter des livres à la bibliothèque', le deux parties de ce topos étant prononcées de manière explicite et reliées par *parce que*.

Ensuite, $U^{(1)}$ réfute l'argument de M via les topoï extrinsèques : 'Plus on a un examen, plus on a besoin du livre 'X'' (voir le paragraphe antérieur dans ce chapitre pour une discussion pourquoi ce topos est extrinsèque), et 'Plus on a besoin d'un certain livre, plus il est probable que le bibliothécaire le prêtera' (ce topos est, lui aussi, extrinsèque parce qu'il se base sur la décision subjective du bibliothécaire de ne pas respecter la règle *conventionnelle* qui, par défaut, limite le nombre des livres qu'un client peut emprunter).

Dans son tour de parole suivant, la machine concède à accepter l'argument de $U^{(1)}$, via un topos extrinsèque 'Plus on est un bon client, plus il est probable que le bibliothécaire prêtera un livre si on le lui demande', qui, ici aussi, est affirmé de manière explicite et marqué par *car*. Cependant, l'argument de $U^{(1)}$ est réfuté dans le cas de $U^{(2)}$, via deux topoï : tout d'abord, un topos intrinsèque dur (une loi naturelle) 'Moins on a d'exemplaires d'un livre, moins on peut emprunter ce livre aux clients' (exprimé comme une règle de la forme 'On ne peut pas prêter

plus de livres qu'on a dans la bibliothèque'), et un topos extrinsèque 'Plus on est en retard dans le retour des livres qu'on avait empruntés, moins il est probable qu'on puisse emprunter des nouveaux livres, avant de retourner les livres précédemment empruntés'. Ce dernier topos (extrinsèque) est utilisé pour biaiser la décision de M vers le prêt du livre à $U^{(1)}$.

On devrait noter que, d'un point de vue discursif, la relation rhétorique de *Consequence* est trouvée valide entre le deuxième et le premier énoncé dans M_1 (c'est-à-dire, les morceaux de texte connectés par *parce que*). Cela est cohérent avec le fait que le topos utilisé dans ce tour de parole est intrinsèque (plus précisément, intrinsèque conventionnel).

Cependant, dans M_2 , le *car* ne marque plus un topos intrinsèque ; de l'autre côté, la relation de *Consequence* n'est pas inférée entre les énoncés reliés par *car*, parce que le fait que 'le bibliothécaire décide de prêter un livre à un client, même si, par défaut, ce dernier n'a plus le droit d'emprunter des livres' n'est pas une conséquence logique du fait que 'ce client est un bon client'.

Le *parce que* dans ce même tour de parole a cependant besoin d'une explication plus subtile : d'un côté, la relation rhétorique de *Consequence* est vraie entre 'mais pas pour vous' et 'je n'ai plus qu'un exemplaire dans la bibliothèque', car le fait que 'le bibliothécaire ne peut emprunter le livre qu'à un seul client' est une conséquence logique du fait que 'il n'y a plus qu'un seul exemplaire empruntable du livre dans la bibliothèque' ; cela est consistant avec le fait que le topos marqué par *parce que* est intrinsèque (plus précisément, intrinsèque dur, comme nous l'avons montré ci-dessus). Néanmoins, le connecteur *et* suggère que 'mais pas pour vous' est aussi une conséquence logique de 'vous n'avez pas encore retourné les livres que vous avez empruntés il y a cinq mois' ; en fait, le premier énoncé résulte du deuxième *via* un topos extrinsèque (comme nous l'avons montré ci-dessus), et *non pas via* une implication logique. Par conséquent, la relation rhétorique de *Consequence* n'est pas vraie entre ces deux énoncés, ce qui est consistant avec le type du topos qui les connecte.

On a ainsi montré comment l'ordonnancement argumentatif des énoncés peut piloter les décisions concernant le choix lexical des connecteurs discursifs, tels que *mais*, *quand même*, *parce que* ou *car*, ce qui est un problème important dans la génération du langage, parce qu'il permet de compléter les mécanismes plutôt rigides, quoique plus rapides que les approches à base de grammaires, de réalisation de surface à base de structures figées, avec des décisions plus dépendantes du contexte. Cependant, dans cette thèse seuls les connecteurs concessifs ont été abordés de manière détaillée. La formalisation des contraintes pour d'autres types de connecteurs, tels que les connecteurs argumentatifs, ou les connecteurs phatiques (tels que *eh ben*, ou le *quoi* en fin d'énoncé), reste un travail de perspective.

8.4 Conclusion et perspectives

Dans ce chapitre nous avons proposé un cadre pour contrôler la génération des connecteurs concessifs afin de mieux structurer les répliques de la machine en dialogue avec les utilisateurs. Le mécanisme proposé pour générer *mais*, *quand même*, *pourtant* et *bien que* s'articule d'une part autour d'une description sémantique de ces connecteurs, dérivée de celle proposée par Moeschler [118] et, d'une autre part, autour d'un ensemble de spécifications pragmatiques dans une procédure computationnelle. Ainsi, l'efficacité de ce cadre a été démontrée sur un frag-

ment typique de dialogue homme-machine autour d'une application particulière – la réservation de documents dans une bibliothèque. Toutefois, les démarches proposées sont génériques, voire indépendantes du domaine d'application et paramétrables par rapport à celui-ci.

Cette recherche devrait être continuée au moins en étendant l'ensemble des connecteurs à d'autres, comme *parce que*, *puisque* (pour cela il faudrait distinguer entre les deux, en utilisant par exemple le concept de *polyphonie* de Bakhtine et Ducrot [55]), *car*, *d'ailleurs*, etc.

De plus, le cadre décrit dans ce chapitre peut également être utilisé à d'autres niveaux dans un système de dialogue, à part le calcul des connecteurs discursifs en génération des tours de parole à plusieurs énoncés en dialogue. Par exemple, en considérant toujours la partie génération d'un système de dialogue homme-machine, le cadre décrit ici peut guider la production des réponses à au moins deux niveaux : (i) à un niveau de surface, nous pourrions guider le choix lexical afin que la force argumentative des énoncés soit réglée plus finement : ainsi, des mots comme 'peu' / 'un peu', 'seulement', 'même', 'et même', etc. (c'est-à-dire, qui n'ont pas de contrepartie logique immédiate) peuvent être insérés de manière appropriée dans les énoncés, afin d'améliorer leur caractère naturel et leur efficacité argumentative ; (ii) à un niveau prosodique, sans affecter la forme logique qui exprime l'intention communicationnelle, ou la forme de surface d'un énoncé.

Chapitre 9

Conclusions et perspectives

9.1 Bilan

9.1.1 Conclusions générales

Lors de notre travail de thèse, nous avons étudié la manière dont ce que les interlocuteurs *disent ouvertement* en dialogue limite le choix des formes linguistiques possibles pour les tours de parole suivants. Afin d'« opérationnaliser » cette étude, nous avons dû effectuer quelques choix méthodologiques :

- (i) tout d'abord, nous avons considéré le *discours* comme une bonne approximation de ce que les interlocuteurs se disent, par conséquent nous avons étudié les diverses tentatives de formaliser cette entité (le discours) ;
- (ii) nous nous sommes limités à quelques phénomènes linguistiques qui pourraient être influencés par le discours, notamment :
 - (a) la réalisation des ellipses dans les énoncés (cf. chapitre 6),
 - (b) la variation dans le degré de puissance des énoncés sur leurs destinataires (c'est ce qu'on appelle le « degré de force illocutoire » des énoncés – cf. chapitre 7 et annexe A),
 - (c) la capacité de décider qu'un énoncé est un argument plus fort qu'un autre, par rapport à une conclusion explicite ou implicite, ciblée par les interlocuteurs et le choix, le cas échéant, du mot connecteur le plus approprié entre deux énoncés (cf. chapitre 8) ;
- (iii) après une première itération sur l'étape (ii), nous nous sommes rendus compte que les faits auxquels les locuteurs s'engagent *publiquement* sont aussi importants que la structuration rhétorique pour contraindre la forme linguistique des énoncés (voir notamment le cas des ellipses dites « sémantiques » – cf. chapitre 6, et le cas du degré de force illocutoire – cf. chapitre 7), par conséquent nous nous sommes décidés de les intégrer à notre cadre.

Ainsi, nous pouvons affirmer que, à la différence des autres tentatives similaires à la nôtre, de modéliser les contraintes sur les énoncés en génération dialogique, nous avons évité explicitement de s'appuyer sur des présupposés concernant les intentions des interlocuteurs, en préférant

de rester agnostiques à l'égard de ces dernières, pour ne prendre en compte que ce que ces locuteurs *disent* effectivement dans leurs interactions. Ce faisant, nous adoptons, à notre avis, une perspective plus modeste sur ce qu'on peut « deviner » dans les *dirés* – comme le disait Ducrot [57] – des interlocuteurs. Cette « modestie » nous permet une généralité accrue des traitements, parce que ces derniers ne dépendent plus explicitement des particularités (voire profils) des interlocuteurs, et cela devient, bien évidemment, intéressant dans le contexte pratique des systèmes de dialogue orienté service, où on interagit avec une éventail potentiellement infini d'utilisateurs humains.

Nous pouvons donc affirmer que cette thèse a essayé de saisir une toute petite partie du passage de ce qu'on souhaite communiquer, vers ce qu'on communique (ou plutôt, on fait sortir) vers les interlocuteurs.

Nous voudrions aussi souligner que le cadre formel proposé se distingue d'autres théories très prisées à l'heure actuelle par certaines caractéristiques ; ainsi :

- à la différence de la Théorie de l'argumentation de Ducrot et Anscombe [6], [58], notre cadre est *formel*, c'est-à-dire, chaque principe est exprimé en termes mathématiques et algorithmiques, ce qui ouvre la voie vers des extensions contrôlées et vers des implémentations dans des systèmes réels ;
- à la différence de la SDRT de Asher et Lascarides, et des travaux, dans le même ordre d'idées mais en génération, de Danlos et Roussarie [149], notre approche est plutôt *procédurale* que déclarative, ce qui, à nouveau, permet d'arriver, nous croyons, plus rapidement à une implémentation pratique ; de plus, également à la différence de ces travaux, nous utilisons *le même* formalisme logique d'un bout à l'autre des traitements, ce qui réduit la complexité de la formalisation, tout en rendant les éventuelles révisions plus contrôlables ; il y a, certes, des inconvénients avec cette approche, notamment ceux que la SDRT essaye d'éviter par un pluralisme de formalismes logiques : par exemple, en SDRT la logique de l'attachement (la « glue ») est même moins complexe qu'une logique du premier ordre, tandis que la logique utilisée pour exprimer les contenus propositionnels est beaucoup plus complexe, car dynamique ;
- à la différence des approches à base d'engagements sociaux en dialogue, comme par exemple celle de Kibble [91], nous intégrons aussi les contraintes d'ordre discursif sur les décisions concernant les énoncés à produire par la machine en dialogue.

Pour revenir à la deuxième distinction méthodologique entre notre cadre et la SDRT (concernant le pluralisme des formalismes logiques), nous soulignons que, même si nous n'utilisons qu'une logique du premier ordre à tous les niveaux, nous arrivons à contourner sa semi-décidabilité en contraignant l'espace des raisonnements au niveau de l'ontologie de tâches : pour un domaine particulier, pour une tâche particulière, il y a un nombre fini de chemins dans cette ontologie, par conséquent les raisonnements que nous pouvons effectuer à partir de cette dernière peuvent être contrôlés.

Avant de clôturer ce paragraphe de conclusions, il faut noter un problème subtil avec notre approche, qui a conduit à l'investigation d'une possible solution : il s'agit de la correspondance entre les valeurs pour le degré de force illocutoire pour un énoncé, et les diverses formes linguistiques pour cet énoncé : une manière évidente de contourner ce problème (et préférée par ailleurs dans des systèmes tels que celui de Stent [165] ou de Theune [169]) est d'annoter, en

termes des degrés de force illocutoire, un ensemble (plutôt volumineux) de structures linguistiques figées, pour coller ensuite ces structures aux énoncés, selon le degré de force souhaité. Une telle approche risque à tout moment d'être contestée comme *ad-hoc*, c'est pourquoi nous avons tenté de trouver une autre solution à ce problème. Il s'agit de partir d'un autre développement récent de Ducrot, la Théorie des modificateurs déréalisants [58] : en prenant comme point de départ la taxinomie et la sémantique (informelle) proposée par Ducrot pour ces modificateurs, nous avons établi un cadre formel où ces modificateurs agissent, de manière *compositionnelle* sur la force de l'énoncé où ils apparaissent. Mais la question est : de quel type de force s'agit-il, d'une force argumentative ou illocutoire ? Ducrot propose la première réponse, puisque pour lui tout passe par l'argumentatif lorsqu'il s'agit de décrire le sens des énoncés ; de toute manière, nous pensons qu'il s'agit plutôt de la force illocutoire. A partir de cette réflexion, nous avons amorcé un travail (encore en état préliminaire, parce que nous n'arrivons pas à traiter toutes les situations possibles – c'est-à-dire, toutes les manières dont une forme linguistique peut réaliser une certaine force illocutoire) qui consisterait à modifier la forme logique de l'énoncé, en lui ajoutant des prédicats (logiques) correspondants à ces modificateurs (déréalisants ou réalisants) pour ainsi varier le degré de force illocutoire de l'énoncé. Ces recherches, dans leur état actuel, sont présentées dans l'annexe A.

9.1.2 Situations délicates en dialogue à plusieurs locuteurs

La plupart des limitations de l'approche basée sur la pragmatique pour contrôler la production des énoncés par la machine, en dialogue avec plusieurs locuteurs, ont leur origine dans un certain « manque de robustesse » par rapport aux caractéristiques des utilisateurs qui pourraient interagir avec le système. Plus précisément, le cadre présenté dans ce chapitre suppose que le système interagit avec un utilisateur « typique », c'est-à-dire *coopératif* (c'est-à-dire, qui agit afin de réaliser un certain but, avec le support du système et pas contre ce dernier) et *rationnel* (c'est-à-dire, il agit d'une manière motivée du point de vue logique, en « calculant » donc des conséquences logiquement valides des affirmations faites) [35]. Cependant, si le système doit être déployé en tant que service (par exemple, un bibliothécaire virtuel), alors il pourrait avoir devant lui une pléthore d'utilisateurs « non-typiques », qui sont (ou deviennent, à un certain moment dans le déroulement du dialogue) non-coopératifs, soit intentionnellement (par ironie ou mauvaise volonté), soit non-intentionnellement (par une attaque soudaine pendant la conversation, ou à cause de différents facteurs psychologiques). Dans ces cas-là, nous pouvons facilement s'imaginer des situations de dialogue où les énoncés de l'utilisateur sont soit trop ambigus pour être interprétés de manière appropriée (lorsqu'une interprétation erronée des tours de parole de l'utilisateur amène la machine à produire un énoncé inadéquat), soit mal-placés ou absents (de manière plus probable, pour des usagers non-coopératifs pour des raisons non-intentionnelles). Ces limitations, concernant la variabilité des utilisateurs potentiels, sont indépendantes du fait que le système travaille dans des situations de bi- ou multi-locuteurs.

Une autre limite intéressante de l'approche concerne une situation de dialogue à plusieurs locuteurs, qui est assez particulière (quoique plutôt fréquente en pratique), que nous appelons « chevauchement séquentiel des tours de parole », où un énoncé est réalisé, de manière séquentielle, par deux ou plusieurs locuteurs. Dans ce cas-là, il n'est pas évident comment la machine pourrait interpréter les contributions (langagières) des usagers, afin de mettre à jour leurs listes

d'engagements ou piles de prédicats. Nous considérons un exemple dans ce sens, où il y a deux utilisateurs, $U^{(1)}$ et $U^{(2)}$, qui se trouvent face au système de dialogue, instancié à son tour dans un bibliothécaire virtuel ; les deux utilisateurs essaient de réserver des livres dans la bibliothèque (les indices inférieurs droits désignent les tours de parole) :

$U_1^{(1)}$: Avez-vous aimé le roman de Coelho...

$U_1^{(2)}$: « L'alchimiste » ? Oui, certainement !

Nous observons qu'il y a un énoncé qui est produit de manière séquentielle et conjointe par deux locuteurs. Le problème, pour la machine, est comment mettre à jour les listes d'engagements de $U^{(1)}$ et $U^{(2)}$. Deux possibilités existent :

1. les listes d'engagements des deux locuteurs sont mises à jour avec le contenu sémantique de la question sur le roman, tandis que le deuxième énoncé, de $U^{(2)}$, est interprété comme étant dans une relation de *QAP* avec le premier énoncé, tout en étant une confirmation de ce premier énoncé ;
2. la liste d'engagements de $U^{(1)}$ n'est pas mise à jour (au cas où $U^{(2)}$ n'a pas « deviné » le livre dont $U^{(1)}$ était en train de parler), tandis que la liste d'engagements de $U^{(2)}$ est mise à jour comme ci-dessus, selon la possibilité antérieure.

Le choix parmi ces possibilités dépend en fait des tours de parole qui *suivent* à celui de $U^{(1)}$; par exemple, $U^{(1)}$ pourrait confirmer l'énoncé de $U^{(2)}$:

$U_2^{(1)}$: Ah, c'est génial, effectivement !

Dans ce cas, les listes d'engagements sont mises à jour selon la situation 1. ci-dessus. De l'autre côté, $U^{(1)}$ pourrait produire un énoncé relié *via* la relation rhétorique de *P-Corr* au deuxième énoncé de $U^{(2)}$, dans son premier tour de parole :

$U_2^{(1)'}$: Non, en fait je voulais dire « Véronique se prépare à mourir »...

Dans ce cas, les listes d'engagements des deux locuteurs sont mises à jour selon la situation 2 ci-dessus. Le problème réside dans le fait que nous avons besoin d'informations *a posteriori* afin de mettre à jour de manière fiable les listes d'engagements, à partir de quelques tours de parole donnés. Une solution apparente pourrait être de garder séparément chaque possibilité (alternative) de mettre à jour les listes d'engagements, dans une structure arborescente (puisqu'il peut souvent y avoir de tels effets de « chevauchement séquentiel », donc des points de décision, dans un dialogue), pour ensuite élaguer cet arbre à chaque fois que la situation nous permet (comme dans l'exemple montré ci-dessus). Cependant, l'élagage pourrait ne pas être possible pour chaque nœud (donc, point de décision) dans l'arbre : si, par exemple, $U^{(1)}$ n'avait rien produit dans son deuxième tour de parole ($U_2^{(1)''}$: ...), alors comment aurait-on pu élaguer l'arbre dans ce point de décision ? Une heuristique pourrait être de supposer que $U^{(1)}$ n'a pas entendu ce que $U^{(2)}$ avait dit ; dans ce cas, nous pouvons élaguer l'arbre selon la situation 2 ci-dessus. Pourtant, s'il n'existe aucun tour de parole de $U^{(1)}$ qui soit relié au tour de parole antérieur de $U^{(2)}$, cela pourrait être interprété comme une confirmation *implicite* par $U^{(1)}$ de l'énoncé de $U^{(2)}$ (voir chapitres 6 et 7). Dans ce cas, une solution pourrait être de s'appuyer sur des tours de parole supplémentaires en dialogue. Par exemple, nous considérons :

$U_2^{(1)''} : \dots$

M_1 : Donc, je mets « L'alchimiste » sur vos cartes, Messieurs ?

$U_3^{(1)}$: Non, en fait je pensais à « Véronique se prépare à mourir » !

Dans ce cas, la liste d'engagements de $U^{(1)}$ est mise à jour selon la situation 1 ci-dessus.

Cependant, à présent cet effet de « chevauchement séquentiel » n'est pas géré par notre cadre ; ici nous avons juste ébauché une manière de le faire.

9.2 Développements futurs

9.2.1 Contournement des limites et relaxation des contraintes

Nous avons montré dans le chapitre 1 (détaillée par la suite tout au long de cette thèse) comment chaque étape dans le contrôle du processus de générer les répliques de la machine en dialogue est un problème de satisfaction de contraintes ; ici, nous allons montrer comment cette idée peut être mise en valeur en étendant la couverture de l'approche (c'est-à-dire, en contournant *quelques-unes* des limites montrées ci-dessus).

Tout d'abord, en ce qui concerne les filtrages locaux pragmatique et sémantique en structuration rhétorique pour le cas du dialogue à deux locuteurs, nous pourrions ajouter des contraintes *douces* sur l'« age » des énoncés précédents qui peuvent être reliés, du point de vue rhétorique, à l'énoncé courant (censé être généré par la machine) : d'autant plus vieux le premier l'est, d'autant plus important le « coût » de sa connexion rhétorique à l'énoncé courant le devient. Ainsi, nous simplifions le calcul de la structure de discours (et aussi les structures mêmes), ce qui est utile pour minimiser le temps de calcul mais, en même temps, nous risquons de perdre la subtilité des représentations rhétoriques des dialogues.

Ensuite, en ce qui concerne le filtrage global, « syntaxique », des relations rhétoriques, nous pourrions transformer les contraintes dures sur l'unification des ensembles de restrictions, en contraintes *douces* : plus important le nombre de variables non-unifiables l'est (c'est-à-dire, les ensembles de restrictions non-unifiables pour ces variables) dans la grammaire abstraite d'attributs associée à une structure de discours (voir chapitre 4), moins la structure de discours est cohérente. Nous pourrions ainsi construire, à chaque mise à jour des structures rhétoriques, une *hiérarchie* de structures de discours incohérentes, à partir de celle la plus incohérente (qui a le nombre le plus réduit de variables globales unifiables), jusqu'à la moins incohérente (qui a toutes les variables globales unifiables). De cette manière, des structures de discours peuvent être calculées pour des dialogues où les énoncés des utilisateurs sont fort ironiques, absurdes (pour des locuteurs avec des problèmes mentaux par exemple), ou pour des dialogues qui ont une cohérence meta-discursive, mais non pas une cohérence rhétorique (voir par exemple le théâtre absurde).

Pour le calcul du degré de force illocutoire, il y a deux possibilités d'agir sur les contraintes impliquées :

- Transformer la contrainte douce de faible distance (de façon idéale, identité) entre les listes d'engagements des interlocuteurs, dans une contrainte *dure*, d'avoir, à tout « moment » en dialogue (c'est-à-dire, après chaque couple de tours de parole entre un usager humain et la machine), les listes d'engagements identiques. Nous sommes ainsi capables

de gérer seulement des dialogues « localement coopératifs » (en opposition avec les dialogues « globalement coopératifs », où un but global est poursuivi par les interlocuteurs et les incidences ou détours sont permis). Un tel dialogue, entre un utilisateur, U et la machine, M , est montré ci-dessous :

M : Vous voulez le livre « Le virtuose » par Margriet de Moor ?

U : Oui, tout à fait. Est-ce que je pourrais avoir aussi le DVD avec le film de Hart, basé sur ce livre ?

M : Oui, biensûr. Je les mets sur votre carte ?

U : Oui, la voilà.

M : OK, je vous remercie, au revoir, bonne journée.

U : De rien, bonne journée à vous aussi !

Dans ce dialogue où une composante multimodale impliquant des gestes est aussi impliquée, il n'y a que des paires de tours de parole dans une relation de *Acknowledgement* (c'est-à-dire, confirmation) l'un avec l'autre et qui réalisent des types d'actes de langage de la forme (F^{FS}, F^{FS}) ou (F^{FS}, F^S) .

- Relâcher la contrainte sur l'opérateur « max » sur les ensembles des degrés de force illocutoire, en dialogue à plusieurs locuteurs, en la remplaçant par des opérateurs de moyenne ou de moyenne partielle (c'est-à-dire, moyenne calculée sur un sous-ensemble des degrés de force pour tous les locuteurs), ou même par l'opérateur « min ». Ainsi, la convergence vers des listes d'engagements identiques pour la machine et pour les utilisateurs est moins abrupte. Cela revient à ce que la machine réalise des contributions moins « marquées ». Par exemple, nous considérons un dialogue entre deux utilisateurs, $U^{(1)}$ et $U^{(2)}$, et la machine, M :

M_1 : Le livre « Le virtuose » de Margriet de Moor se trouve dans le secteur de littérature internationale, aux écrivains néerlandais.

$U_1^{(1)}$: En fait, j'ai déjà cherché, je suis sûr qu'il n'y est pas du tout.

$U_1^{(2)}$: Oui, moi aussi, j'ai regardé là, je crois que je ne l'ai pas vu non plus, mais je ne suis pas sûr...

M_2 : Messieurs, je suis absolument sûre, le livre doit être là, donc je vous prie de regarder de nouveau !

Dans ce dialogue, l'opérateur « max » est appliqué sur les forces illocutoires calculées pour les locuteurs $U^{(1)}$ (+2) et $U^{(2)}$ (+1) (voir chapitre 7). Si l'opérateur « min » avait été appliqué entre ces quantités, la machine aurait généré un énoncé plus « doux » (avec un degré de force de +1) :

M'_2 : Oui, pourtant je pense que le livre est là, donc, il est peut-être mieux que vous regardiez de nouveau, pour être sûrs à cent pourcent...

Nous croyons toutefois qu'un degré maximal d'adéquation (du point de vue du caractère naturel perçu par les usagers) pourrait être atteint si la machine était capable de commuter de manière automatique entre ces opérateurs, c'est-à-dire, entre les contraintes. Malheureusement, à présent nous ne sommes pas en possession d'un ensemble de critères qui puissent guider ce type de décisions.

Enfin, en ce qui concerne le contrôle de l'ellipse sémantique, nous pourrions relâcher la contrainte dure de n'éliminer (dans la forme logique de l'énoncé à générer) que les prédicats contenus dans l'intersection des piles de prédicats pour tous les interlocuteurs, en la remplaçant par une contrainte *douce*, d'éliminer des prédicats contenus dans des intersections **partielles** de piles de prédicats, ou même, dans un cas extrême, que les prédicats contenus dans la **réunion** des piles de prédicats soient éliminés. Dans le premier cas, l'énoncé elliptique serait destiné à un nombre réduit d'interlocuteurs, tandis que dans le deuxième cas, l'énoncé serait destiné à un sujet *omniscient*, qui aurait « enregistré » tout ce que les autres avaient dit (par exemple, une secrétaire personnelle). Ainsi, nous pourrions traiter des dialogues plus ambigus.

Pendant, toutes ces modifications des contraintes devraient être appliquées seulement suite à des tests sur des corpus de dialogue réels, pour ainsi évaluer leur pertinence ; *a priori*, les mécanismes présentés dans cette thèse gèrent les situations de dialogue où les locuteurs sont coopératifs, sincères et rationnels, ce qui représente un point de départ intéressant vers des développements futurs, qui devraient avoir lieu en rapport avec les particularités des tâches traitées par le système et, implicitement, en fonction des caractéristiques des utilisateurs.

9.2.2 Perspectives à moyen et à long terme

Les travaux menés à l'occasion de cette thèse sont ouverts à plusieurs développements futurs. Ainsi, nous envisageons deux catégories de perspectives : à moyen terme et à long terme.

En ce qui concerne les perspectives à moyen terme, il s'agit surtout d'évaluer de manière quantitative, sur des corpus de dialogues multi-locuteurs, le cadre proposé. Pour ce faire, il faut d'abord trouver des corpus de dialogues multi-locuteurs, notamment en français, qui soient disponibles en accès libre. Pour le moment, le problème de trouver des corpus accessibles dans le domaine public reste ouvert. Un autre problème à cet égard concerne également le genre des dialogues : nous croyons que ce dernier peut introduire un biais sur les résultats que nous pourrions obtenir en appliquant notre cadre sur un tel corpus, par conséquent il faut y avoir une adéquation entre le genre du corpus et le genre des dialogues cibles, autour d'une tâche particulière. C'est pourquoi, à la différence de ce que nous avons proposé en [139], nous ne croyons plus que les pièces de théâtre soient une bonne approximation d'un corpus d'évaluation. Il n'en est pas moins que, quel que soit le corpus utilisé, il faut au moins parcourir les étapes suivantes, pour évaluer, de manière indépendante d'autres modules dans un système de dialogue particulier, notre cadre :

1. annotation des énoncés dans le corpus : chaque réplique dialogique produite par un locuteur devrait être annotée en termes de l'acte de langage, du degré de force illocutoire, et de la forme logique, censée exprimer l'intention communicationnelle « derrière » cette réplique¹ ;
2. génération des formes de surface équivalentes : pour chaque paire (acte de langage, formule logique) déterminée à l'étape précédente, nous devrions générer à la main un ensemble *minimal* de formes linguistiques équivalentes, en variant le degré de force illo-

¹La forme logique de chaque réplique peut être obtenue soit à la main, soit à l'aide d'un analyseur sémantique, tel que Phoenix [122], mais dans ce dernier cas il faudrait y avoir une ontologie correspondant au domaine couvert par le corpus.

cutoire, l'ellipse sémantique et aussi les connecteurs entre des actes différents, dans une même réplique ;

3. choix automatique des marqueurs pertinents (ellipse sémantique, degré de force illocutoire et connecteurs), en utilisant notre cadre ;
4. choix automatique de la forme linguistique (parmi celles générées à l'étape 2) qui satisfait les contraintes imposées lors de l'étape 3 ;
5. comparaison entre la forme linguistique choisie à 4 et la forme linguistique d'origine, dans le corpus ; si ces formes avaient été identiques, alors notre cadre aurait bien approximé le comportement dialogique des locuteurs ; sinon, il ne l'aurait pas.

Il s'agit donc d'une évaluation extrinsèque par rapport aux éléments manipulés et contrôlés dans notre cadre, au sens où ce n'est que le résultat final des traitements (c'est-à-dire, le choix de la forme linguistique appropriée) qui compte ; de plus, l'évaluation est cumulative, au sens où le résultat *final* de tous les traitements est évalué par rapport au corpus d'origine, plutôt que chaque traitement partiel ; cela parce que, nous pensons, des limites à un niveau peuvent pallier d'autres limites à un autre niveau.

Cette évaluation pourrait être appliquée à chacune des répliques dans le corpus de dialogues ; à la fin, un score pourrait être calculé, comme un rapport entre le nombre d'énoncés générés « correctement » (c'est-à-dire, qui sont identiques aux énoncés d'origine) et le nombre total des énoncés générés.

En ce qui concerne les perspectives à long terme, celles-ci visent au moins trois directions, que nous allons détailler ci-après :

1. couplage entre le cadre formel proposé ici et un générateur de surface, en faisant attention notamment à la réalisation des différents degrés de force illocutoire ;
2. extension du cadre formel proposé dans cette thèse, afin de contrôler également la prosodie, en dialogue multi-locuteurs ;
3. développement de méthodes pour construire automatiquement des ontologies de tâches, à partir des corpus de dialogues.

Pour la génération de surface, nous envisageons au moins deux pistes parallèles :

- (i) utiliser une grammaire déterministe à base de contraintes, éventuellement réversible (c'est-à-dire, utilisable à la fois pour l'analyse sémantique – le passage du texte vers la représentation logique, pour les énoncés produits par les utilisateurs humains d'un système de dialogue, et pour la génération de surface – le passage de la formule logique vers le texte, la chaîne orthographique correspondante, pour les énoncés que la machine est censée produire en dialogue), comme par exemple celle de Neumann [121].
- (ii) utiliser une approche stochastique où, à partir d'exemples dans un corpus de couverture suffisante, annoté en formules logiques (ou en cadres sémantiques), on apprend des correspondances entre ces formules logiques et les chaînes orthographiques ; ensuite, pour assurer une certaine compositionnalité à l'approche, afin que l'ensemble des énoncés générés soit assez vaste, nous pourrions utiliser un formalisme inspiré des travaux de Bod, avec son modèle d'analyse orientée par les données (en anglais, « Data-Oriented Parsing » – DOP) [23] ; un exemple particulièrement pertinent dans ce sens est constitué par les travaux de Ratnaparkhi [146].

La première approche envisagée a l'avantage évident de garantir une certaine homogénéité par rapport à tout le cadre proposé dans cette thèse, au sens où le même paradigme – la résolution des contraintes – offrirait un cadre unifié et une perspective unitaire sur une partie plus étendue (en incluant aussi la génération des énoncés proprement dits) du passage de l'intention vers la parole. Un deuxième avantage de l'approche réside dans la possibilité d'y greffer facilement le traitement des modificateurs (déréalisants et réalisants) pour faire varier le degré de force illocutoire des énoncés générés (cf. annexe A). Cette voie a toutefois quelques inconvénients importants, notamment un relatif manque de flexibilité concernant les écartements des énoncés oraux par rapport à un ensemble fixé d'avance de règles grammaticales. En effet, pour l'instant il ne nous paraît pas évident comment nous pourrions intégrer des phénomènes tels que les conséquences linguistiques des ellipses sémantiques, dans un cadre tel que celui de Neumann. Par conséquent, cette voie est à explorer plus profondément, pour en déceler finement les limites.

La deuxième approche envisagée permet, à notre avis, de contourner les inconvénients de la première (notamment, les problèmes de flexibilité par rapport au caractère « oral » des énoncés), en ajoutant toutefois quelques problèmes. Ainsi, dans [23], Bod propose d'annoter des corpus en termes de cadres sémantiques associés à chaque mot, syntagme ou expression, pour ensuite combiner ces cadres sémantiques, *via* des règles apprises (de manière probabiliste) sur le corpus ; par conséquent, le formalisme logique qui se prêterait à un tel traitement serait justement une représentation du type cadres sémantiques, ce qui entraînerait la nécessité d'introduire une étape supplémentaire d'interfaçage entre notre cadre formel, tel qu'il a été proposé dans cette thèse, et la génération de surface. Plus précisément, il s'agit d'un niveau de conversion des formules logiques du premier ordre en cadres sémantiques. Ce fait pourrait nous poser des problèmes concernant par exemple le traitement des phénomènes de portée (notamment les quantifications) dans les structures rhétoriques (SDRS) calculées lors de chaque ajout d'un tour de parole à un dialogue. Par conséquent, notre cadre formel devrait être lui-même adapté pour pouvoir travailler avec des cadres sémantiques au lieu des formules logiques, plutôt que de convertir simplement les dernières vers les premiers ; pour ce faire, nous pourrions nous inspirer des travaux de Irmer, qui pour l'instant ne concernent cependant que les situations de monologue [82]. De plus, il nous semble que l'intégration du mécanisme pour passer du degré de force illocutoire à la forme de surface (tel qu'il est proposé dans l'annexe A), dans un cadre inspiré du modèle DOP, n'est pas évidente.

En ce qui concerne le calcul de la prosodie, cela est très important pour atteindre un degré d'efficacité suffisant pour les tours de parole générés par la machine en dialogue. Pour coupler par exemple les relations rhétoriques de la SDRT avec des phénomènes prosodiques tels que l'accent (« stress ») des mots, en interprétation de certains types d'énoncés, comme les questions du type 'Il fait beau aujourd'hui, n'est pas ?' (en anglais, les « tag questions »), il y a des études, comme celle de Asher *et al.* [15], qui proposent de traiter ces marques prosodiques comme des indices sur l'intentionnalité du locuteur, ou sur certaines contraintes discursives (comme par exemple le changement de topique) – voir notamment l'étude de Jayez et Dargnat sur les élèvements continuatifs (en anglais, « continuative rise » – CR) de la fréquence fondamentale en fin d'énoncé [85]. Dans cette dernière étude, après avoir d'abord utilisé ces CR comme des marqueurs sur le manque de changement de topique (donc comme une sorte d'éléments pour l'établissement de ponts – « bridging » – entre des énoncés différents), ils parviennent à offrir

une interprétation de ce phénomène en termes des croyances de l'interlocuteur, ce qui s'éloigne de l'esprit de notre cadre formel, de rester agnostiques à l'égard des états intentionnels des partenaires de dialogue.

L'étude de Jayez et Dargnat reste inspiratrice pour nous, dans la mesure où il suffirait de renverser le sens des règles qu'ils proposent, pour utiliser leurs conséquences, résultées de l'interprétation des CR, comme des préconditions pour la génération de ces CR. Cependant, un problème, qui apparaît par ailleurs aussi dans l'approche de Asher *et al.* [15], est que l'analyse se déroule au niveau de l'énoncé en entier ; pour pouvoir contrôler ces contours prosodiques, surtout en génération, il faudrait une granularité plus fine de l'analyse. Pour ce faire, nous envisageons une piste qui, par ailleurs, n'est pas tout-à-fait nouvelle dans ses éléments fondamentaux : il s'agit de « descendre » à l'intérieur de l'énoncé, pour individualiser les parties nouvelles (le rhème) des parties déjà situées dans l'arrière-plan des locuteurs (le thème). Pour arriver à opérationnaliser cette distinction, nous avons besoin tout d'abord de la forme textuelle des énoncés dont la prosodie est à calculer, pour ensuite les synthétiser ; ensuite, nous avons besoin d'un mécanisme pour réaliser cette séparation entre thème et rhème ou, selon la terminologie de l'Ecole linguistique de Prague, entre topique et focus. Pour analyser à ce degré de finesse les énoncés, cela tout en prenant en compte les contraintes discursives (exprimées selon le formalisme de la DRT – « Discourse Representation Theory » de Kamp et Reyle [90]), Korbayová a proposé un cadre unifié pour intégrer l'idée d'état informationnel aux DRS (« Discourse Representation Structure ») correspondant à un discours (monologique) [95]. Ainsi, l'auteur de ces travaux arrive à établir un pont entre la DRT et la Description générative fonctionnelle (en anglais, « Functional Generative Description » – FGD) de l'Ecole de Prague [158] pour ainsi parvenir à analyser plus finement que dans la DRT d'origine, des phénomènes comme les anaphores associatives dans les textes monologiques.

Après avoir proposé cette extension de la DRT avec la FGD, Korbayová s'est mise à travailler dans le projet européen SIRIDUS (<http://www.ling.gu.se/projekt/siridus>) pour essayer d'appliquer son cadre théorique [95] au calcul de la prosodie des énoncés en dialogue bi-locuteurs [96]. Ainsi, elle arrive à montrer (avec les autres partenaires dans le projet) que l'insertion de l'état informationnel de l'énoncé dans la DRS correspondante peut effectivement offrir une analyse assez fine pour permettre de contrôler (voire moduler) le contour de la fréquence fondamentale, par des patrons d'accentuation au niveau de chaque mot.

Cependant, l'approche proposée par Korbayová *et al.* a des limites, concernant notamment le manque d'une représentation sur les relations (rhétoriques) entre les énoncés, ce qui est particulièrement important pour le dialogue. Autrement dit, l'approche de ces auteurs n'a pas encore été, à notre connaissance, étendue à la SDRT, même si cela est reconnu comme une nécessité même dans l'étude initiale de Korbayová en 1998 [95]. De plus, la prise en compte des contraintes que les situations de dialogue multi-locuteurs imposent sur les patrons d'accentuation reste encore un problème ouvert. Ainsi, pour s'attaquer à ces problèmes, nous envisageons d'adopter comme point de départ ces travaux de Korbayová *et al.* [96], pour les étendre dans un premier temps avec des relations rhétoriques entre les énoncés, dans un contexte de dialogue bi-locuteurs, et enfin élargir ce cadre au moins aux situations de dialogue multi-locuteurs analysés dans le chapitre 5 de cette thèse.

Enfin, une des limites *pratiques* les plus importantes de notre cadre formel (comme par

ailleurs aussi de la SDRT, par exemple) réside dans la nécessité d'avoir une base de connaissances encyclopédiques pour pouvoir inférer les relations (notamment causales) entre les contenus sémantiques des énoncés, et cela lors de la mise à jour de la structure rhétorique (cf. notamment chapitre 3), ou lors du calcul des topoï argumentatifs (cf. chapitre 8). Dans l'état courant du développement de ce cadre, plusieurs telles bases de connaissances (structurées sous forme d'ontologies) ont été construites à la main, pour des domaines particuliers, tels que la réservation des salles dans un laboratoire de recherches, ou la réservation et l'emprunt d'un ensemble très limité (une dizaine) de livres dans une bibliothèque (hypothétique, puisqu'aucune bibliothèque de taille acceptable ne possède qu'une dizaine de livres !). La construction manuelle de ces ontologies s'avère toutefois être un processus très coûteux dans le temps ; c'est pourquoi il faudrait y avoir une procédure qui permette de construire automatiquement une telle ontologie, à partir d'un corpus de dialogues pertinents pour une tâche particulière. Ainsi, nous avons déjà amorcé des travaux préliminaires à cet égard où nous avons intersecté l'ensemble des concepts dans le WordNet, avec les concepts extraits d'un corpus (acquis *via* la méthode du magicien d'Oz) de dialogues bi-locuteurs autour de la réservation de salles dans un laboratoire de recherche. Ensuite, le graphe obtenu en extrayant du WordNet les concepts présents aussi dans ce corpus, avec les relations entre eux (en tant que précisées dans l'ontologie de départ), nous avons obtenu une sorte de « sous-ontologie » de WordNet. Ces travaux ont été implémentés dans une étude menée par Hagiescu et nous-mêmes [72] et évalués en comparant l'ontologie ainsi extraite avec celle construite au préalable, à la main.

Suite à cette étude, nous nous sommes rendus compte (comme par ailleurs nous nous attendions) qu'une telle ontologie n'était pas suffisante pour nos buts, puisque dans le WordNet il n'y a pas de relations syntagmatiques entre les concepts. C'est pour cette raison qu'il faut, nous pensons, poursuivre d'abord une méthodologie similaire à celle de Tufiş *et al.* dans la construction du BalkaNet [183], en essayant de *porter*, à partir d'un corpus représentatif pour une tâche particulière, le WordNet vers une ontologie lexicalisée particulière, où les éventuelles inconsistances (par exemples, des cycles dans l'arbre paradigmatique des concepts) devraient être corrigées à la main. Dans un deuxième temps, ce « mini-WordNet » obtenu pour la tâche devrait aussi être augmenté avec des relations syntagmatiques entre les concepts, notamment celles qui établissent des relations non-hiérarchiques entre les concepts (comme par exemple le fait que Sophocle soit l'auteur de « Oedipe roi »). Pour ce faire, une possibilité serait d'intersecter d'abord la hiérarchie de ce mini-WordNet (tributaire, nous le soulignons, de la hiérarchie contenue dans le WordNet) avec la hiérarchie d'une autre base de connaissances, à vrai dire encyclopédiques, comme par exemple la Wikipédia. Des recherches à cet égard ont été déjà amorcées par des scientifiques comme Veale, qui essaye justement de combiner les différentes sources de connaissances, pour modéliser des phénomènes comme la métaphore [186] ; par conséquent, ces démarches pourraient être adaptées aux dialogues ciblés par notre cadre, parce que si, comme nous l'avons vu le long de cette thèse, le système agit aussi comme un conseiller pour les utilisateurs, alors il faudra être capable d'effectuer des raisonnements sur les contenus des livres.

Chacune de ces perspectives représente un sujet de développement (pour l'évaluation sur des corpus de dialogues) ou de recherche à part, qui dépasse, à notre avis, le cadre d'une seule thèse. En fait, chacune des perspectives à long terme pourrait être abordée dans une thèse à part. De plus, nous pensons qu'il est bien de séparer conceptuellement les tâches de contrôle

« profond » de la génération (ce qui a été abordé dans cette thèse), de réalisation de surface et de réglage prosodique, pour mieux mettre en évidence comment différentes facettes du même contexte (pragmatique) d'interaction sont mises en valeur pour chacune des étapes sur le chemin de l'intention communicationnelle vers la parole. Dans cette thèse seule la première étape, à notre avis la moins étudiée à présent, a été abordée.

Annexe A

Matérialisation du degré de force illocutoire au niveau de la forme de surface – étude préliminaire

A.1 Introduction

Dans le chapitre 7 nous avons proposé un cadre formel pour calculer le degré de la force illocutoire réalisée par un énoncé, spécifié sous forme logique. Cependant, nous avons laissé de côté le problème de mettre en correspondance ces consignes concernant la force illocutoire avec la forme de surface des énoncés ; en fait, nous avons contourné ce problème en utilisant des structures linguistiques figées, annotées d'avance en termes de leur degré de force illocutoire. Dans cette annexe nous proposons une façon possible de contrôler cette correspondance de manière plus rigoureuse et théoriquement justifiée, quoique partielle ; partielle, au sens où le cadre que nous proposons ici ne modélise pas toutes les correspondances possibles entre le degré de force et la forme de surface.

Avant de décrire en détail l'approche proposée, nous précisons qu'il s'agit essentiellement de mettre en valeur un des travaux récents de Ducrot, sur les modificateurs *déréalisants* [58] : nous stipulons qu'en ajoutant des modificateurs déréalisants ou réalisants à certaines entités dans les représentations « syntaxiques » correspondantes aux formes logiques des énoncés (cf. chapitre 8) – notamment au prédicat et à l'objet de l'énoncé – nous pouvons varier de manière décroissante ou, respectivement, croissante, le degré de force illocutoire de l'énoncé concerné. Pour formaliser cette idée, nous nous écartons tout de même des idées de Ducrot, au sens où nous considérons que les phénomènes liés à la variation du degré de puissance d'un énoncé en rapport avec la présence des modificateurs déréalisants ou réalisants, tiennent plutôt à la force illocutoire qu'à la force argumentative, comme le prétend Ducrot [58].

Ainsi, dans le paragraphe suivant nous dressons un bilan critique des propos de Ducrot concernant les modificateurs déréalisants et réalisants, pour ensuite proposer une formalisation de ces propos, en insistant notamment sur la *compositionnalité* des modificateurs, à la fois au niveau de la même entité (prédicat ou objet) dans un énoncé, et au niveau de l'énoncé entier

(en prenant donc en compte les effets des modificateurs appliqués à toutes les entités concernées dans un énoncé – le prédicat et l’objet). Enfin, nous reprenons une partie d’un exemple étendu de dialogue à plusieurs locuteurs, introduit dans le chapitre 5 et réutilisé dans les chapitres 6 et 8, pour montrer les avantages, mais aussi les limites du cadre proposé, dans son état actuel, pour réaliser la correspondance entre les valeurs du degré de force illocutoire et les diverses formes linguistiques pour un énoncé précisé sous forme sémantique.

A.2 Les modificateurs déréalisants

Dans [58] Ducrot continue les travaux de Anscombe sur les topoï intrinsèques et extrinsèques (voir [6], mais cf. aussi chapitre 8), au sens où il propose un moyen d’ordonner les topoï intrinsèques selon leur force argumentative. Pour ce faire, il s’appuie sur la notion de modificateurs *réalisants* et *déréalisants*. Selon Ducrot, un modificateur déréalisant d’un mot w est défini comme un mot μ tel que le syntagme $w\mu$ a soit une orientation argumentative inverse, soit une force argumentative moindre, que le mot w . Tout autre modificateur du mot w (c’est-à-dire, avec la même orientation argumentative que w et tel que $w\mu$ a une force argumentative égale ou supérieure à w) est dénommé modificateur réalisant de w ¹. Un autre aspect important, pour les modificateurs réalisants et déréalisants μ , est que le syntagme $w\mu$ n’est pas senti comme contradictoire [58]². De plus, Ducrot montre qu’un critère pour détecter les modificateurs déréalisants (appelés désormais MD) est la non-acceptabilité d’un énoncé du type ‘ w et même $w\mu$ ’, sauf au cas où un contexte factuel (extrinsèque) est disponible et / ou le locuteur a une intention argumentative complexe (c’est-à-dire, il se base sur des connaissances extrinsèques aux mots en tant que tels). Ensuite, Ducrot montre que les MD peuvent agir soit comme des atténuateurs, c’est-à-dire ils réduisent la force argumentative d’un énoncé, soit comme des inverseurs, c’est-à-dire ils changent l’orientation argumentative de l’énoncé (plus précisément, du mot auquel le modificateur est appliqué). Un exemple classique est constitué par le couple *peu / un peu*. L’importance de ces modificateurs, selon Ducrot, réside dans le fait qu’ils induisent une relation d’ordre argumentatif sur les items lexicaux (et donc sur les topoï qui utilisent ces items). À partir de cela, nous pouvons déduire que deux énoncés qui impliquent la même conclusion et ne diffèrent qu’à l’égard d’un modificateur appliqué à un mot (un verbe ou un substantif) peuvent être ordonnés selon leur force argumentative (à l’égard d’une certaine conclusion).

À la différence de Ducrot, nous proposons que ces modificateurs induisent en effet une relation d’ordre sur les énoncés, mais que cette relation n’est pas argumentative, mais illocutoire. Pour voir que cela est le cas, il faut d’abord préciser que, lorsqu’il définit les notions de modificateurs réalisants et déréalisants, Ducrot généralise la notion de degrés comparatifs appliqués aux adjectifs et aux adverbes : les modificateurs réalisants généralisent l’idée de PLUS, tandis que les modificateurs déréalisants généralisent l’idée de MOINS. Ces notions peuvent être assimilées aux prédicats logiques, même si Ducrot, lui-même, ne les voit pas de cette manière. Par exemple, au lieu de n’avoir que ‘Pierre a plus avancé que Jean’ / ‘Pierre a moins avancé que

¹Il faut noter la différence entre les connecteurs « soit » et « et » entre les contraintes qui définissent les modificateurs déréalisants et, respectivement, réalisants.

²Cette contrainte est plus légère que le principe de non-contradiction de la sémantique véri-conditionnelle : ici il suffit que le syntagme ne soit pas *senti* comme contradictoire, par celui qui la verbalise.

Jean', il peut y avoir aussi 'Pierre a avancé rapidement' / 'Pierre a avancé lentement' ; l'idée est que 'avancer rapidement' est *plus* 'avancer' qu' 'avancer', aussi qu' 'avancer lentement' est *moins* 'avancer' qu' 'avancer'. De plus, il existe des items lexicaux (verbes, adjectifs) qui n'ont pas de degré comparatif : on ne peut pas dire 'Pierre est plus parti que Jean'. Cependant, selon Ducrot, les verbes comme 'partir' sont graduels eux aussi : selon cet auteur, l'énoncé 'Pierre est parti tôt' est comme 'Pierre est *plus* parti', donc l'adverbe 'tôt' attribue le degré PLUS au verbe 'partir'.

Un autre point dans la théorie de Ducrot sur les modificateurs est la distinction entre les MD *atténuateurs* et *inverseurs*. Les atténuateurs sont des MD qui ne font que réduire la force (argumentative selon Ducrot, illocutoire selon nous) des mots auxquels ils sont appliqués, cela tout en gardant la même orientation de cette force (on ne voit pas un équivalent illocutoire de la notion d'orientation argumentative). D'un point de vue linguistique, les modificateurs peuvent être soit des adjectifs (lorsqu'ils sont appliqués à un substantif), soit des adverbes (lorsqu'ils sont appliqués à un verbe). Ducrot prétend que, dans certaines langues (dont le français) lorsqu'un adjectif déréalisant est, du point de vue syntaxique, un « attribut » (en fait, un nom prédicatif), il est toujours inverseur. Ainsi, l'énoncé 'Le changement de Paris est lent ; tu ne seras pas dépaysé' est parfaitement cohérent, parce que la conclusion de 'Paris change' est 'on sera dépaysé', mais l'ajout de 'lent' change d'orientation pour entraîner la conclusion 'on ne sera pas dépaysé'. De l'autre part, lorsque le MD est un « épithète » dans la terminologie de Ducrot (en fait, un attribut syntaxique), il ne modifie pas l'orientation argumentative d'un substantif auquel il est appliqué ; par exemple, dans l'énoncé 'Il y a eu de lents changements à Paris ; tu seras peut-être dépaysé', le mot 'peut-être' a été ajouté afin de mieux accommoder l'effet d'atténuation que 'lent' a sur 'changement'. D'un autre côté, bien évidemment, un adjectif réalisant ne peut jamais être un inverseur, qu'il soit en position de nom prédicatif ou d'attribut. D'un point de vue « formel » nous verrons que la différence entre ces deux fonctions des adjectifs peut être exprimée comme l'adjectif étant un argument de l'agent de l'énoncé (le cas du nom prédicatif), ou de l'objet de l'énoncé (le cas de l'attribut).

Comme nous le verrons plus loin, pour l'instant nous n'utiliserons que les MD atténuateurs (ou les MR) pour diminuer (ou, respectivement, augmenter) le degré de force illocutoire des énoncés ; pour les MD inverseurs, il n'y a pas d'utilisation illocutoire évidente, donc ce type de modificateurs (MD en tant qu'argument de l'agent de l'énoncé) ne sera pas utilisé pour essayer de mettre en correspondance les degrés de force illocutoire avec les formes linguistiques.

A.3 Formalisation

Pour formaliser dans un cadre logique les MD et MR, il faut d'abord observer que, selon la théorie de Ducrot, les modificateurs (potentiellement déréalisants) ne déterminent que des verbes (prédicats au niveau logique) et des substantifs (objets, agents ou patients au niveau logique). Il existe donc une correspondance assez évidente entre les adverbes et adjectifs réalisants ou déréalisants, et nos modificateurs de la forme $[\text{mod}]_{\langle p \rangle}$ et, respectivement, $[\text{mod}]_{\langle o \rangle}$ (cf. chapitre 8 pour une discussion sur les niveaux de représentation des énoncés). La manière la plus évidente d'ordonner, d'un point de vue illocutoire, les entités du type $\langle \text{pred} \rangle$ versus $\langle \text{pred} \rangle \uparrow [\text{mod}]_{\langle p \rangle}$, et $\langle \text{obj} \rangle$ versus $\langle \text{obj} \rangle \uparrow [\text{mod}]_{\langle o \rangle}$, est de construire à la main un tableau de cor-

respondances, contenant, pour chaque concept dans l'ontologie de tâches, un ensemble de modificateurs réalisants et déréalisants ; par exemple, pour le concept PARENT, nous avons l'entrée PARENT :: CLOSE₊; DISTANT₋, où + indique un MR et - indique un MD. Un tel ensemble d'entrées est construit à la main au moment de la conception d'une application particulière, par un linguiste qui peut utiliser les tests de Ducrot avec *mais* et *et même* (voir le paragraphe antérieur, mais cf. aussi [58] pour une discussion plus détaillée).

Cependant, ce que nous pouvons faire automatiquement, pour chaque énoncé, est déterminer si un certain MD est atténuateur ou inverseur, cela pour voir si des effets illocutoires peuvent lui être associés. Pour ce faire, nous utilisons la forme syntaxique (déterminée selon les principes du chapitre 8, à partir de la forme logique fournie par le contrôleur de dialogue) de l'énoncé, et les critères de Ducrot : si le MD est un adjectif qui est réalisé au niveau syntaxique comme un nom prédicatif, alors il est inverseur ; inversement, si l'adjectif déréalisant a la fonction syntaxique d'attribut, alors il est atténuateur. Un adjectif réalisé comme un nom prédicatif revient à une égalité entre l'agent et l'entité réalisée, morphologiquement, comme le substantif déterminé par cet adjectif (qui est un prédicat du type *feature*) ; donc, dans ce cas, l'agent est identique à l'objet de l'énoncé :

$$([\text{agt}] = v_1) \uparrow (\langle \text{pred} \rangle = v_2) \uparrow ([\text{mod}]_{\langle p \rangle} = m_i)^* \uparrow ([\text{obj}] = v_1) \uparrow ([\text{mod}]_{[\text{o}]} = m_j)^* \uparrow ([\text{pat}] = v_3).$$

Un tel énoncé est 'Les changements en Astronomie ont été lents dans les derniers deux siècles'³.

Un adjectif qui est attribut du point de vue syntaxique est réalisé ainsi qu'il détermine une entité qui est l'objet de l'énoncé (par exemple, 'Je n'aime pas ces lents changements que l'Astronomie a subis dans les deux derniers siècles !') ; d'un point de vue formel, cela revient à une forme syntaxique de l'énoncé (libellé π) comme :

$$([\text{agt}] = \text{emitter}(\pi)) \uparrow (\langle \text{pred} \rangle = \text{'dislike'}) \uparrow ([\text{obj}] = \text{'modif'}) \uparrow ([\text{mod}]_{[\text{o}]} = \text{'small'}) \uparrow \dots$$

Cependant, au cas où les modificateurs sont des adverbes (c'est-à-dire, de la forme $[\text{mod}]_{\langle p \rangle}$), nous supposons *par défaut* que tous les MD (tels que spécifiés dans le tableau de correspondances) sont des atténuateurs ; bien évidemment, cela représente une limitation de notre traitement, dans son état actuel.

Un aspect essentiel que tout cadre sémantique doit traiter d'une manière ou d'une autre est la *compositionnalité* : au niveau des recherches décrites dans cette annexe, cela revient à décider comment ordonner (du point de vue illocutoire) les énoncés lorsque plusieurs modificateurs (MD ou MR) apparaissent. Il y a deux manières dont plusieurs modificateurs peuvent apparaître dans un énoncé :

(i) à une entité sémantique nous appliquons plusieurs modificateurs :

$$\{\langle \text{pred} \rangle | [\text{obj}]\} \uparrow (\uparrow_{i \geq 1} [\text{mod}]_{\langle p \rangle | [\text{o}]}^{(i)});$$

(ii) des modificateurs sont appliqués à plusieurs entités sémantiques dans un énoncé :

$$\langle \text{pred} \rangle \uparrow [\text{mod}]_{\langle p \rangle} \uparrow [\text{obj}] \uparrow [\text{mod}]_{[\text{o}]}.$$

Dans le premier cas, nous rappelons que chaque modificateur peut être soit un MR (cela est marqué par $([\text{mod}]_{\langle p \rangle | [\text{o}]}^{(i)} = m_i^{(+)})$), soit un MD ; chaque MD peut être soit un atténuateur (cela est marqué par $([\text{mod}]_{\langle p \rangle | [\text{o}]}^{(i)} = m_i^{(-)})$), soit un inverseur (cela est marqué par $([\text{mod}]_{\langle p \rangle | [\text{o}]}^{(i)} = -m_i)$).

³Ici 'Astronomie' est le modificateur de l'agent (c'est-à-dire, de 'changements'), et le prédicat est 'ont été lents' (c'est-à-dire, formellement, $\text{small}(X) \wedge \text{equals}(X, \text{'modif'})$).

Nous stipulons que l'opération de concaténation des modificateurs est associative, c'est-à-dire, pour une entité w et deux modificateurs μ_1 et μ_2 , nous avons :

$$w\mu_1\mu_2 := w \uparrow \mu_1 \uparrow \mu_2 = (w \uparrow \mu_1) \uparrow \mu_2 = w \uparrow (\mu_1 \uparrow \mu_2).$$

Ensuite, en utilisant l'associativité de l'opération de concaténation, nous exprimons l'effet de plusieurs modificateurs sur une entité *via* une fonction de « norme », définie, pour tout modificateur μ_i , ainsi :

$$\|\mu_i\| = \begin{cases} \exp(1) & \text{si } \mu_i = m_i^{(+)}, \\ \exp(-1) & \text{si } \mu_i = m_i^{(-)}, \\ -1 & \text{si } \mu_i = -m_i, \\ 1 & \text{si } \mu_i = \emptyset \end{cases}$$

Ensuite, l'effet (réalisant, atténuateur ou inverseur) d'un ensemble de modificateurs $\uparrow_{i \geq 1} \mu_i$ sur une entité w est donné par :

$$\|\uparrow_{i \geq 1} \mu_i\| = \prod_{i \geq 1} \|\mu_i\| = (-1)^{|\{\mu_i; \mu_i = -m_i\}|} \cdot \exp(\sum_{i \geq 1; \mu_i \neq -m_i} \ln(\|\mu_i\|)),$$

où $|A|$ désigne le nombre d'éléments dans l'ensemble A ⁴.

Considérons par exemple la construction 'Des changements ont été faits' ; l'objet et l'agent sont instanciés par 'changements' ($[\text{agt}] = \text{'change'}$) \uparrow ($[\text{obj}] = \text{'change'}$). Nous ajoutons ensuite le MD 'marginal' : 'Des changements marginaux ont été faits' ; dans ce cas l'objet est déréalisé ($[\text{obj}] = \text{'change'}$) \uparrow ($[\text{mod}]_{[\text{obj}]} = m_{\text{marginal}}^{(-)}$). Nous ajoutons ensuite le MR 'important' : 'Des changements marginaux importants ont été faits' ; dans ce cas l'objet déréalisé est réalisé ($[\text{obj}] = \text{'change'}$) \uparrow ($[\text{mod}]_{[\text{obj}]}^{(1)} = m_{\text{marginal}}^{(-)}$) \uparrow ($[\text{mod}]_{[\text{obj}]}^{(2)} = m_{\text{important}}^{(+)}$) ; l'orientation argumentative du syntagme 'changements marginaux importants' par rapport à 'changements' est donnée par $(-1)^0 \cdot \exp((-1) + (+1)) = 1$, ce qui est équivalent au fait que 'changements marginaux importants' a la même force *illocutoire* que 'changements', pour *n'importe quelle* conclusion que l'énoncé pourrait déterminer⁵. Nous ajoutons ensuite le MD 'peu (de)' : 'Peu de changements marginaux importants ont été faits' ; dans ce cas la force de 'peu de changements marginaux importants' (c'est-à-dire, syntaxiquement, $[\text{obj}] = \text{'change'}$) \uparrow ($[\text{mod}]_{[\text{obj}]}^{(1)} = m_{\text{marginal}}^{(-)}$) \uparrow ($[\text{mod}]_{[\text{obj}]}^{(2)} = m_{\text{important}}^{(+)}$) \uparrow ($[\text{mod}]_{[\text{obj}]}^{(3)} = m_{\text{a few}}^{(-)}$) par rapport à 'changements' est donnée en multipliant la « norme » de $[\text{mod}]_{[\text{obj}]}^{(3)}$ par l'effet antérieur de 'marginal important' sur 'changement' : $\|m_{\text{a few}}^{(-)}\| \cdot 1 = \exp(-1)$; donc, l'effet de 'peu de ... marginaux importants' sur 'changements' est de déréalisation, ce qui veut dire que la dernière construction composée a un degré de force illocutoire moindre par rapport à l'entité initiale, sans attributs ('changement').

En ce qui concerne le deuxième type de compositionnalité, celle où il y a plusieurs entités modifiées dans un énoncé, nous adoptons tout d'abord l'heuristique d'attribuer une priorité plus haute aux prédicats : autrement dit, s'il y a un énoncé dont nous voulons varier le degré de force illocutoire, nous appliquons d'abord des MD ou MR à son prédicat et, dans un deuxième temps, à son objet, lorsqu'il n'y a plus de modificateurs disponibles, applicables au prédicat.

⁴L'utilisation de l'opérateur produit entre les « normes » peut être motivée par un parallèle avec la théorie des probabilités, si nous considérons que les événements du type 'un modificateur est appliqué à une entité' sont, *a priori*, mutuellement indépendants.

⁵Bien évidemment, cela n'est pas toujours le cas dans l'utilisation réelle du langage, puisque si, *via* des marqueurs prosodiques, nous accentuons de manière plus prégnante 'important', alors le syntagme 'changements marginaux importants' est légèrement réalisée. A notre avis, ces aspects ne sont pas, pour l'instant, formalisables dans notre cadre.

Quelques exemples aideront à clarifier ces points ; ainsi, considérons le même énoncé « souche » :

(0) Des changements ont été faits.

Ensuite, considérons les énoncés « dérivés » :

(1) Beaucoup de changements ont été faits.

(2) Peu de changements ont été faits.

(3) Des changements ont été faits lentement.

(4) Des changements ont été faits rapidement.

(5) Beaucoup de changements ont été faits lentement.

(6) Peu de changements ont été faits lentement.

(7) Peu de changements ont été faits rapidement.

(8) Beaucoup de changements ont été faits rapidement.

Ces énoncés peuvent être représentés de manière plus condensée en soulignant le caractère réalisant ou déréalisant des modificateurs : (1') '+ de changements ont été faits', (2') '- de changements ont été faits', (3') 'Des changements ont été - faits', (4') 'Des changements ont été + faits', (5') '+ de changements ont été - faits', (6') '- de changements ont été - faits', (7') '- de changements ont été + faits', (8') '+ de changements ont été + faits'. Dans ces exemples, l'agent coïncide avec l'objet (ils sont tous les deux instanciés par 'changements'), tandis que le prédicat est instancié par 'ont été faits' ; par conséquent, les modificateurs de l'agent / objet sont des adjectifs à fonction attributive⁶, tandis que les modificateurs du prédicat sont des adverbes à fonction syntaxique complétive. Selon notre cadre (et le principe de compositionnalité précisé ci-dessus), ces huit énoncés sont ordonnés, du point de vue de leur force illocutoire, comme suit :

(6) < (2) < (7) < (3) < (0) < (4) < (5) < (1) < (8).

Evidemment, un premier problème avec ce cadre réside dans le manque apparent d'appariement entre cet ordonnancement des énoncés et l'échelle à cinq niveaux pour les degrés de force illocutoire (cf. chapitre 7). Ainsi, pour l'instant nous proposons l'heuristique suivante : à l'énoncé « souche » (c'est-à-dire, sans modifiants réalisants ou déréalisants) nous attribuons le degré de force illocutoire de 0 (zéro) ; ensuite, aux énoncés où l'effet global des modificateurs appliqués au prédicat est réalisant et l'effet global des modificateurs appliqués à l'objet est déréalisant nous attribuons le degré +1, aux énoncés où l'effet global des modificateurs appliqués au prédicat est déréalisant et l'effet global des modificateurs appliqués à l'objet est réalisant nous attribuons le degré -1, aux énoncés où les effets globaux sur le prédicat et l'objet sont réalisants nous attribuons le degré +2 et, enfin, aux énoncés où les effets globaux sur le prédicat et l'objet sont déréalisants, nous attribuons le degré -2. Nous pouvons exprimer de manière concise ce principe, sous forme de cinq règles, ainsi :

1. $\pi : \dots \uparrow (\langle \text{pred}^\pi \rangle = \dots) \uparrow ([\text{mod}^\pi]_{\langle p \rangle} = \emptyset) \uparrow ([\text{obj}]^\pi = \dots) \uparrow ([\text{mod}^\pi]_{[o]} = \emptyset) \Rightarrow \partial\phi((\dots, \pi)) = 0 ;$

⁶On peut ainsi voir que ni la réalisation syntaxique d'un adjectif comme un nom prédicatif dans un énoncé n'implique forcément l'égalité entre l'agent et l'objet de l'énoncé, ni inversement.

TAB. A.1 – Degrés de force illocutoire pour un ensemble d'énoncés

Énoncé	Degré de force illocutoire
Des changements ont été faits.	0
Beaucoup de changements ont été faits.	+2
Peu de changements ont été faits.	-2
Des changements ont été faits lentement.	-2
Des changements ont été faits rapidement.	+2
Beaucoup de changements ont été faits lentement.	-1
Peu de changements ont été faits lentement.	-2
Peu de changements ont été faits rapidement.	+1
Beaucoup de changements ont été faits rapidement.	+2

2. $\pi : \dots \uparrow (\langle \text{pred}^\pi \rangle = \dots) \uparrow ([\text{mod}^\pi]_{\langle p \rangle} = m^{(+)}) \uparrow ([\text{obj}]^\pi = \dots) \uparrow ([\text{mod}^\pi]_{[o]} = m^{(-)}) \Rightarrow \partial\phi(\dots, \pi) = +1$;
3. $\pi : \dots \uparrow (\langle \text{pred}^\pi \rangle = \dots) \uparrow ([\text{mod}^\pi]_{\langle p \rangle} = m^{(-)}) \uparrow ([\text{obj}]^\pi = \dots) \uparrow ([\text{mod}^\pi]_{[o]} = m^{(+)}) \Rightarrow \partial\phi(\dots, \pi) = -1$;
4. $\pi : \dots \uparrow (\langle \text{pred}^\pi \rangle = \dots) \uparrow ([\text{mod}^\pi]_{\langle p \rangle} = m^{(+)}|\emptyset) \uparrow ([\text{obj}]^\pi = \dots) \uparrow ([\text{mod}^\pi]_{[o]} = m^{(+)}|\emptyset) \wedge \neg([\text{mod}^\pi]_{\langle p \rangle} = \emptyset) \wedge ([\text{mod}^\pi]_{[o]} = \emptyset) \Rightarrow \partial\phi(\dots, \pi) = +2$;
5. $\pi : \dots \uparrow (\langle \text{pred}^\pi \rangle = \dots) \uparrow ([\text{mod}^\pi]_{\langle p \rangle} = m^{(-)}|\emptyset) \uparrow ([\text{obj}]^\pi = \dots) \uparrow ([\text{mod}^\pi]_{[o]} = m^{(-)}|\emptyset) \wedge \neg([\text{mod}^\pi]_{\langle p \rangle} = \emptyset) \wedge ([\text{mod}^\pi]_{[o]} = \emptyset) \Rightarrow \partial\phi(\dots, \pi) = -2$.

Tandis que les premières trois règles sont claires, car elles ne font que formaliser ce que nous avons précisé ci-dessus, les deux dernières règles méritent plus de discussion ; tout d'abord, nous observons que dans les contraintes concernant les valeurs des modificateurs nous avons un choix entre une valeur non-vide ($m^{(+)}$ ou $m^{(-)}$) et une valeur vide, non-attribuée (\emptyset). Cela parce qu'il suffit que l'effet *résultant* des modificateurs du prédicat et de l'objet soit réalisant ou déréalissant, lorsque les effets séparés des modificateurs sur le prédicat et sur l'objet ne sont pas contradictoires. De l'autre côté, l'effet résultant ne peut être vide (car si cela était le cas, nous nous retrouverions dans la situation couverte par la règle 1), donc nous ne pouvons pas avoir les modificateurs du prédicat et de l'objet à effet vide en même temps. D'autre part, si les modificateurs sur le prédicat et sur l'objet ont des effets contraires, cela revient aux situations couvertes par les règles 2 et 3, la résultante globale de ces effets ne peut pas être vide, puisque les prédicats sont prioritaires par rapport aux objets.

Ainsi, étant donné le même énoncé « souche » présenté ci-dessus, celui-ci a un degré de force neutre (zéro). En revanche, les degrés de force illocutoire pour les autres énoncés sont spécifiés dans le tableau A.1.

Evidemment, ce mécanisme ne recouvre pas toutes les possibilités de réaliser les forces illocutoires ; il ne s'agit que d'une étude préliminaire qui a essayé d'explorer la possibilité de mettre en valeur la notion de modificateur (réalisant et déréalissant) pour mettre en correspondance le degré de force illocutoire avec la forme de surface, et cela d'une autre manière que *via* des structures linguistiques figées annotées. Peut-être que ce serait plus réaliste de combiner les deux approches, car, par exemple, en n'utilisant que ces modificateurs, nous ne pourrions jamais pro-

duire des formes de surface telles que celles montrées dans le chapitre 7. Dans le paragraphe suivant de cette annexe nous allons toutefois essayer d’aller plus loin sur cette voie (utilisation des modificateurs) pour voir comment, étant données une formule logique (qui correspond à un énoncé) et une valeur pour le degré de force illocutoire, des modificateurs réalisants ou déréalisants peuvent être *insérés* dans cette formule logique, afin de permettre d’arriver à une forme de surface qui reflète ce degré de force particulier.

A.4 Génération superficielle des degrés de force illocutoire

La force illocutoire d’un énoncé qui n’est disponible que sous forme logique peut être « propagée » au niveau superficiel en modifiant cette formule logique, c’est-à-dire, en y ajoutant des *modificateurs*, appliqués aux prédicats et aux objets de cet énoncé. Cela est réalisé dans notre cadre en ajoutant des conjoints de la forme $\text{feature}(\langle \text{pred} \rangle, \text{Modp})$, ou $\text{feature}(\langle \text{obj} \rangle, \text{Modo})$ où $\langle \text{pred} \rangle$ et $\langle \text{obj} \rangle$ sont le prédicat et, respectivement, l’objet, et Modp et Modo sont des *variables*, liées aux valeurs des modificateurs du prédicat et, respectivement, de l’objet. Le cadre compositionnel que nous avons présenté dans le paragraphe précédent offre plusieurs possibilités de combiner les modificateurs pour varier le degré de force illocutoire (comme par ailleurs nous l’avons vu dans le paragraphe antérieur). Maintenant, nous allons voir comment nous pouvons, à partir d’une formule logique pour un énoncé à générer, y ajouter des modificateurs, afin que l’énoncé ait le degré de force illocutoire souhaité, et cela au niveau de surface.

Considérons par exemple l’énoncé ‘Il n’y a pas de tragédie ancienne écrite par Aristophane ou Virgile où un inceste apparaît’, inspiré de l’exemple de dialogue discuté dans les chapitres 5 et 6 ; cet énoncé, qui a une force illocutoire neutre (de degré zéro), peut avoir son degré de force illocutoire augmenté en ajoutant des modificateurs à son prédicat ($\text{write}/2$) et / ou à son objet ($\text{ancient_greek_tragedy}$). Ainsi, la formule logique pour cet énoncé est⁷ :

$$\text{FS}(\exists X \forall Y, Z : \text{agent}(X) \wedge \text{equals}(X, \emptyset) \wedge \text{object}(Y) \wedge \text{equals}(Y, \text{ancient_greek_tragedy}) \cdots \wedge \text{write}(Y, Z) \wedge \text{equals}(Z, \text{author}) \wedge \neg(\text{equals}(Z, \text{aristophanes}) \vee \text{equals}(Z, \text{vergilius})))$$

Si nous voulons que cette forme logique soit réalisée au niveau linguistique avec un degré de force illocutoire donné, il faut ajouter des modificateurs (réalisants ou déréalisants) au prédicat et / ou à l’objet de cet énoncé. Cependant, en modifiant l’objet (\neg ’tragedy’, donné par la construction niée sur les caractéristiques – *feature* – de l’objet) avec un MR tel que ‘une seule’ nous amène à obtenir un énoncé plus puissant, au niveau illocutoire, que l’original (cela parce que le modificateur ‘une seule’, même s’il est déréalisant pour un objet spécifié de manière positive – ‘tragedy’ – il est réalisant pour un objet spécifié de manière négative – \neg ’tragedy’.) Ainsi, dans ce cas nous obtenons un énoncé avec un degré de force illocutoire +2 (le modificateur ajouté a été souligné) :

Il n’y a pas une seule tragédie ancienne écrite par Aristophane ou Virgile où un inceste apparaît.

⁷Nous avons simplifié la présentation de cette forme logique, en en omettant des informations non pertinentes pour la discussion menée ici, notamment les informations sur le temps de l’action précisée dans cet énoncé, ou encore sur des éléments de détail concernant l’objet (comme par exemple le fait qu’il s’agit d’une tragédie où un inceste apparaît).

Cette modification sur la forme de surface de l'énoncé est déterminée par des modifications sur la forme logique de l'énoncé, en y ajoutant un modificateur sur l'objet ; ainsi, la nouvelle formule logique, à partir de laquelle nous obtenons l'énoncé montré précédemment, est (nous avons marqués en gras les ajouts à cette forme logique) :

$$F^S(\exists X, T, U \forall Y, Z : \text{agent}(X) \wedge \text{equals}(X, \emptyset) \wedge \text{object}(Y) \wedge \text{equals}(Y, \text{object}^{(\pi_{22})}) \wedge \text{feature}(Y, T) \wedge \text{count}(T) \wedge \text{feature}(T, U) \wedge \text{equals}(U, 1) \wedge \text{write}(Y, Z) \wedge \text{equals}(Z, \text{'author'}) \wedge \neg(\text{equals}(Z, \text{'aristophanes'}) \vee \text{equals}(Z, \text{'vergilius'})))$$

Le prédicat *count/1* attribue un type à son argument, c'est-à-dire, il est vrai lorsque son argument est un nombre qui compte un ensemble d'entités. Du fait que l'énoncé a un degré de force illocutoire très fort selon le cadre ébauché ici, le fait d'ajouter des modificateurs réalisants au prédicat par exemple, n'augmentera pas le degré de force. Nous pourrions également produire un énoncé avec un degré de force illocutoire de -1 , en ajoutant un modificateur déréalisant au prédicat, pour obtenir un énoncé tel que (les ajouts sont soulignés) :

Il n'y a pas une seule tragédie ancienne probablement écrite par Aristophane ou Virgile où un inceste apparaît.

Evidemment, il est nécessaire que le générateur de surface réalise un réordonnement des mots dans les énoncés, comme par exemple dans le cas de l'énoncé montré ci-dessus : en effet, une forme de surface plus naturelle avec un degré de force de -1 serait 'Il n'y a probablement pas une seule tragédie ancienne écrite par Aristophane ou Virgile, où un inceste apparaît'. Mais cela reste un sujet pour des recherches futures.

Bien évidemment, le mécanisme proposé dans cette annexe est loin de gérer toutes les relations possibles entre les forces illocutoires et les formes de surface ; de plus, ce que nous avons obtenu à présent semble s'éloigner dans une certaine mesure des intuitions dégagées à partir des exemples donnés dans le chapitre 7, concernant la matérialisation linguistique du degré de force illocutoire. Il n'en est pas moins, à notre avis, que ce cadre peut constituer une piste fructueuse vers une correspondance moins *ad-hoc* entre le degré de force illocutoire et la forme linguistique des énoncés.

Annexe B

Glossaire

Ici on va expliciter et définir un ensemble de termes importants pour la compréhension des démarches décrites le long de cette thèse ; les définitions présentées sont celles acceptées dans la communauté des linguistes et donc inspirées du document de référence [54]. Ceci étant, on présente ci-dessous la liste alphabétique de ces concepts, accompagnés des définitions correspondantes.

A

acte

1. On appelle *acte de parole* l'énoncé effectivement réalisé par un locuteur déterminé dans une situation donnée.

2. On appelle *acte de langage* l'utilisation du langage comme une action, et non plus seulement comme un message.

A l'origine, *acte de langage* traduit l'anglais « speech act », donc l'association de l'acte de langage soit à la langue soit à la parole n'est pas évidente. La problématique du précurseur des actes de langage, Austin, n'est pas essentiellement linguistique. Ce dernier part de l'observation des performatifs tels que 'promettre' ou 'baptiser' qui, employés à la bonne personne et dans les circonstances sociales opportunes, réalisent *l'acte qu'ils dénomment*. 'Je te baptise Paul' affecte effectivement un nom à une personne si les *conditions de succès* sont réussies. Dès lors, la problématique s'étend à tout acte de langage : dans un énoncé comme 'Ferme la fenêtre !' on distingue l'*acte locutoire* (prononcer cette phrase), l'*acte illocutoire* (donner un ordre) et l'*acte perlocutoire* (viser à obtenir que la fenêtre soit fermée). L'observation des actes de langage amène à considérer aussi des actes de langage *indirects* où la visée perlocutoire n'est pas associable à une forme illocutoire codée dans la langue. Le social (présent dans l'illocutoire) est ici reporté au perlocutoire : 'Il fait froid dans cette chambre' constitue un acte de langage indirect (équivalent à l'acte perlocutoire 'Ferme cette fenêtre !') si la situation comporte un locuteur susceptible de donner un ordre à l'interlocuteur et d'être obéi.

analyse de discours

On appelle *analyse de discours* la partie de la linguistique qui détermine les règles commandant la production des suites de phrases structurées.

L'analyse de discours trouve son origine dans la distinction (faite par de Saussure) entre la langue et la parole, qui a déterminé le maintien du courant d'une linguistique de la parole opposant à la fonction de communication, essentielle pour l'étude de la langue, une fonction d'expression (phénomènes émotionnels, subjectifs, individuels) qui pose les problèmes de l'étude des énoncés supérieurs à la phrase, notamment de tout ce qui touche à l'énonciation.

La linguistique essaye de rendre compte non seulement de la phrase, mais aussi des suites de phrases à partir de trois séries de travaux ; les uns tentent de définir les règles qui commandent la succession des signifiés d'un texte : c'est l'analyse de contenu. De leur côté, les lexicologues, après avoir longtemps pris comme base opérationnelle le mot, ou unité graphique isolée, ont reconnu la nécessité de prendre en compte des environnements plus larges (co-occurrences, incompatibilités, oppositions) ; l'unité lexicale est ainsi réinsérée dans le tissu des phrases qui la contiennent et son étude implique la référence au discours dans lequel elle apparaît (d'où la nécessité d'une typologie des discours). De même, les linguistes structuralistes insistent sur l'idée de sous-codes propres à chacune des fonctions du langage. Enfin, les linguistes américains des années '50-'60, qui ne se préoccupent ni des fonctions du langage, ni de la distinction théorique entre langue et parole, admettent que la linguistique doit s'attacher également aux énoncés supérieurs à la phrase, et ils élaborent (avec Harris), au moyen des classes d'équivalence, une véritable procédure de réduction et de formalisation du discours ; cela a été le début des approches formelles aux discours, continuées ensuite avec des démarches telles que celles de Kamp et Reyle, Mann et Thompson, Asher et Lascarides, Sperber et Wilson, etc.

L'analyse de discours a pour sujet essentiel la relation du sujet parlant au processus de production des phrases (énonciation) ou la relation du discours au groupe social à qui il est destiné (sociolinguistique). L'analyse de discours proprement linguistique se fonde sur des opérations de réduction de phrases et sur certains concepts de la grammaire transformationnelle. Selon Harris, on établit un certain nombre de classes d'équivalence (découpage du texte en éléments apparaissant dans des contextes identiques) ; mais on est conduit à admettre que seule la connaissance des règles de la langue permet de « normaliser les énoncés » et de réduire, par exemple, 'Des centaines de livres ont été empruntés' à 'On a emprunté des centaines de livres', ces opérations permettant d'établir plus rapidement les classes d'équivalence. Le discours résulte de l'application des règles à un certain nombre de phrases de base : la structure du discours et son histoire transformationnelle.

Dans ce cas, la production des énoncés est étudiée à partir des propositions de base qui aboutissent à des phrases de surface (phrases réalisées) ; le mode de passage de la phrase profonde à la phrase de surface (appelé aussi *génération du discours en langue naturelle*) est considéré comme l'indice d'un comportement linguistique qu'on met ensuite en parallèle avec un comportement non-linguistique (mais psycho-social, politique, culturel, etc.).

analyse conversationnelle

L'*analyse conversationnelle* étudie les règles sous-jacentes au fonctionnement des conversations et des échanges communicatifs, ces règles socio-culturelles variant d'une société à l'autre.

L'analyse conversationnelle s'insère dans le champ plus large de la linguistique de l'interaction qui étudie, dans une perspective fondamentalement dialogale, des unités supérieures à la phrase. Ce dépassement des unités traditionnelles de la linguistique conduit à une démarche trans-disciplinaire.

En effet, l'analyse conversationnelle, issue principalement de l'ethnométhodologie (Sacks) et de la pragmatique, intéresse également l'analyse de discours, la psycho-sociologie, la sociologie interactionnelle et l'ethnographie de la communication. On peut définir au moins trois axes de recherche : (i) l'étude des prises de parole dans le contexte de la production (génération) ; (ii) l'étude des compétences individuelles et du contexte social dans la gestion des échanges langagiers ; (iii) la mise en évidence des multiples niveaux du traitement des données dans l'échange verbal.

C

communication

La *communication*, dans le sens utilisé tout au long de cette thèse, est l'échange verbal entre un sujet parlant, qui produit un énoncé destiné à un autre sujet parlant, et un interlocuteur dont il sollicite l'écoute et / ou une réponse explicite ou implicite (selon le type d'énoncé). La communication est intersubjective. Sur le plan psycho-linguistique, c'est le processus au cours duquel la signification qu'un locuteur associe aux sons est la même que celle que l'auditeur associe à ces mêmes sons.

Les *participants à la communication*, ou acteurs de la communication, sont les « personnes » : l'ego, ou sujet parlant qui produit l'énoncé, l'interlocuteur ou allocutaire, enfin ce dont on parle, les êtres ou objets du monde.

La *situation de communication* est définie par les participants à la communication, dont le rôle est déterminé par le « je » (ego), centre de l'énonciation, ainsi que par les dimensions spatio-temporelles de l'énoncé ou contexte situationnel : relations temporelles entre le moment de l'énonciation et le moment de l'énoncé (les aspects et les temps), relations spatiales entre le sujet et les objets de l'énoncé, présents ou absents, proches ou éloignés, relations sociales entre les participants à la communication ainsi qu'entre eux-mêmes et l'objet de l'énoncé (les types de discours, les facteurs historiques, sociologiques, etc.).

Le *statut de la communication* est défini par la distance sociale, ou intersubjective, instituée par « je » avec ses interlocuteurs (ainsi la différence entre « tu » et « vous » traduit une intimité ou une relation sociale différente), et par la manière dont « je » envisage son énoncé. Ainsi, l'énoncé peut être plus ou moins pris en considération, assumé par le locuteur : cela se traduit par les modes et les aspects du verbe, et par les adverbes, tels que 'peut-être' ou 'sans doute'.

compétence

La compétence est le système de règles interiorisé par les sujets parlants et constituant leur savoir linguistique, grâce auquel ils sont capables de prononcer ou de comprendre un nombre infini de phrases inédites. La compétence d'un sujet parlant une certaine langue explique la possibilité qu'il a de construire, reconnaître et comprendre les phrases grammaticales, de déceler et d'interpréter les phrases ambiguës, de produire des phrases nouvelles. Cette intériorisation de

la grammaire rend compte aussi de l'intuition du sujet parlant, c'est-à-dire, de la possibilité qu'a ce dernier de porter un jugement sur la grammaticalité des énoncés présentés, de décider si une suite quelconque de mots ou caractères appartient ou non à la langue, de repérer les paraphrases. Une des tâches de la linguistique est de définir cette compétence, commune aux locuteurs qui appartiennent à une même communauté linguistique. La compétence s'oppose à la performance, définie par l'ensemble des contraintes qui s'exercent sur la compétence pour en limiter l'usage : la performance, qui correspond partiellement à la « parole », rend compte des utilisations diverses de la langue dans les actes de parole (voir plus loin).

La compétence est le résultat d'une abstraction et d'une idéalisation des données linguistiques directement accessibles à l'observation, qui, elles, appartiennent au domaine de la performance : il s'agit des actes de parole individuels, des textes, discours, etc., qui non seulement représentent l'actualisation du système de compétence, mais sont également tributaires de nombreux paramètres liés à la situation et aux dispositions psychologiques des sujets (mémoire, attention, niveau émotionnel, etc.). Ainsi, ce n'est que lorsqu'on dispose des hypothèses suffisantes sur le système, c'est-à-dire la grammaire, qu'on peut se préoccuper de savoir comment les locuteurs l'utilisent dans des situations concrètes ; c'est pourquoi l'étude de la compétence précède légèrement celle de la performance.

D

discours

1. Le *discours* est le langage mis en action, la langue assumée par le sujet parlant.

2. Le *discours* est une unité égale ou supérieure à la phrase ; il est constitué par une suite formant un message ayant un commencement et une clôture.

3. En rhétorique, le *discours* est une suite de développements oratoires destinés à persuader ou à émouvoir et structurés selon des règles précises. On distingue le genre démonstratif (blâme ou louage), le genre délibératif (conseil ou dissuasion), le genre judiciaire (défense ou accusation). Le discours rhétorique se compose de six parties qui n'entrent pas toutes nécessairement dans un discours : (i) l'exorde, (ii) la proposition ou division (exposé du sujet), (iii) la narration (exposition des faits), (iv) la preuve ou confirmation (moyens sur lesquels on s'appuie), (v) la réfutation (rejet des objections), (vi) la péroraison (conclusion qui persuade et émeut). Les discours relèvent aussi des circonstances dans lesquelles ils sont prononcés : éloquence de la chaire (sermon), du barreau (plaidoyer, réquisitoire), éloquence académique (commémoration).

4. Dans son acception linguistique moderne, le terme de *discours* désigne tout énoncé supérieur à la phrase, considéré du point de vue des règles d'enchaînement des suites de phrases.

Dans la problématique antérieure à l'analyse de discours, le terme de discours ne pouvait être, du point de vue linguistique, que synonyme d'*énoncé*. L'opposition énoncé / discours marquait simplement l'opposition entre linguistique et non-linguistique. La linguistique opérait sur les énoncés qui, regroupés en corpus, s'offraient à l'analyse ; les règles du discours, c'est-à-dire l'étude des processus discursifs justifiant l'enchaînement des suites de phrases, étaient renvoyées à d'autres modèles et à d'autres méthodes, en particulier à toute perspective qui prendrait en compte le sujet parlant.

La conception de l'énoncé comme (partie de) discours demande que soient formulées les règles d'enchaînement, les processus discursifs.

La première tentative dans ce sens est celle de Harris, traitant les phrases comme les unités justiciables d'un traitement comparable à celui auquel la linguistique « traditionnelle » soumet les autres unités de la langue. Les discours présentent des traits formels caractéristiques. Toutefois, comme, à la différence de ce qui se passe pour les unités d'un rang inférieur (morphèmes ou phonèmes, par exemple), l'identité de deux environnements est rare au niveau de la phrase, on doit d'abord définir la notion d'équivalence : le rapprochement des deux énoncés 'Ici les feuilles tombent vers le milieu de l'automne' et 'Ici les feuilles tombent vers la fin du mois d'octobre' permet d'affirmer l'équivalence linguistique entre 'le milieu de l'automne' et 'la fin du mois d'octobre', car l'environnement est identique. L'analyse fonde donc sur ce principe la recherche des classes d'équivalence ; le travail, regroupé dans des tableaux, fait apparaître les récurrences de classes dans le texte.

Toutefois, le discours tel qu'il est défini ci-dessus ne saurait se contenter de ce type d'analyse. Si les procédures de classification sont légitimes, il reste à rendre compte du modèle de communication justifiant les processus discursifs. Jakobson et Benveniste, par la considération des fonctions du langage, modifient le concept de parole ; par les marques de l'énonciation, le sujet parlant ordonne la langue en fonction de « je » et « tu ». L'analyse de discours, réintroduisant le sujet d'énonciation, nécessite l'appel fait à la psycho-linguistique et à la socio-linguistique.

E

énoncé

Le mot *énoncé* désigne toute suite finie de mots d'une langue émise par un ou plusieurs locuteurs. La clôture de l'énoncé est assurée par une période de silence avant et après la suite de mots, silences réalisés par les sujets parlants. Un énoncé peut être formé d'une ou plusieurs phrases ; on peut parler d'énoncé grammatical ou agrammatical, sémantique ou asémantique. On peut adjoindre à un énoncé un adjectif qualifiant le type de discours (énoncé colloquial, littéraire, polémique, etc.), le type de communication (énoncé parlé ou écrit), le type de langue (énoncé français, anglais, etc.). Un ensemble d'énoncés constitue les données empiriques (corpus) d'analyse ; selon la théorie utilisée, cette analyse expliquera les énoncés produits ou prédira les énoncés possibles au regard des règles régissant les phrases dans ce corpus.

énonciation

L'*énonciation* est l'acte individuel de production, dans un contexte déterminé, ayant pour résultat un énoncé ; les deux termes s'opposent comme la *fabrication* s'oppose à l'objet *fabriqué*. L'énonciation est l'acte individuel d'utilisation de la langue, alors que l'énoncé est le résultat de cet acte, c'est l'acte de création du sujet parlant devenu alors *sujet de l'énonciation*. Il s'agit essentiellement, pour les initiateurs de ce concept (notamment Austin et Searle, entre autres), de dégager les éléments qui, dans les énoncés, peuvent être considérés comme les traces ou les empreintes des *procès d'énonciation* qui les ont produits, puis de dégager leur fonctionnement, leur organisation, leur interaction.

I**illocutoire**

A la suite de Austin, on qualifie d'*illocutoire* tout acte de langage réalisant ou tendant à réaliser l'action dénommée : la phrase 'Je promets de ne plus fumer' réalise l'acte de « promettre » en même temps qu'il indique la nature de la promesse. On distingue notamment, parmi les verbes illocutoires, les verbes performatifs (tels que 'ordonner') et les verbes d'attitude (tels que 'jurer'). Tout énoncé, pratiquement, peut être considéré d'une manière ou d'une autre comme (ayant une composante) illocutoire.

L**langagier**

On utilise couramment, depuis les années '80, l'adjectif *langagier*, à côté de l'adjectif *linguistique*. Passer du linguistique au langagier, c'est prendre en compte le *processus* de production des discours en intégrant les paramètres situationnels et humains. *Linguistique* renvoie au principe d'immanence qui consiste à étudier la langue comme formant un ordre propre, autonome, dont il est possible de décrire les structures par leurs seules relations. *Langagier* renvoie au principe de réalité par lequel la structure va se confronter à des besoins communicatifs, des enjeux discursifs, des représentations sociales.

On parle de *compétence linguistique* pour désigner l'ensemble des moyens en langue à la disposition des sujets. On parlera de *compétence langagière* lorsqu'il s'agira de l'utilisation de ces moyens linguistiques dans des situations réelles d'interlocution où il faudra convaincre, se défendre, rassurer, faire semblant, s'imposer à autrui, expliquer. Cette compétence (langagière) se manifeste aussi dans la compréhension comme dans la production (génération) du langage, du discours, c'est-à-dire à l'occasion de la construction du sens.

P**performance**

La *performance* est la manifestation de la compétence des sujets parlants dans leurs multiples actes de parole.

Les performances linguistiques des sujets parlants sont des phrases réalisées dans les situations diverses de communication ; elles forment les données observables qui constituent le corpus d'analyse linguistique.

La performance dépend de la compétence (le système des règles) du sujet parlant, de la situation de communication ; elle dépend, en effet, de facteurs très divers, comme la mémoire, l'attention, le contexte social, les relations psycho-sociales entre le locuteur et l'interlocuteur, l'affectivité des participants à la communication, etc.

perlocutoire

On donne le nom de *perlocutoires* aux fonctions du langage qui ne sont pas inscrites directement dans l'énoncé, mais qui ont un effet indirect sur l'interlocuteur (flatter, faire plaisir, faire peur, etc.). Par exemple, une interrogation peut avoir pour objet non pas d'obtenir une

information, mais de faire croire à l'interlocuteur qu'on le fait participer à la décision (fausse interrogation).

pragmatique

Sous le nom de *pragmatique* on regroupe des orientations très diverses. A l'origine, elle a concerné les caractéristiques de l'utilisation du langage (motivations psychologiques des locuteurs, réactions des interlocuteurs, types socialisés de discours, objet du discours, etc.) par opposition à l'aspect syntaxique et sémantique de la communication langagière. Ensuite, avec l'étude des actes de langage, la pragmatique s'est étendue aux modalités d'assertion, à l'énonciation et au discours pour englober les conditions de succès et de satisfaction des actes de langage et l'analyse conversationnelle (cf. la discussion au début du premier chapitre de cette thèse).

R

rhétorique

On appelle *rhétorique* l'ensemble des procédés constituant l'art oratoire, l'art du bien-dire. La rhétorique comporte trois composantes essentielles : (i) l'*invention* (thèmes et arguments), (ii) la *disposition* (arrangement des parties, structuration des relations entre les parties et les énoncés) et surtout (iii) l'*élocution* (choix et disposition des mots) ; on y ajoute parfois la (iv) *prononciation* (mode d'énonciation) et la (v) *mémoire* (ou mémorisation). L'élocution, objet principal de la rhétorique, se définit essentiellement par l'étude des figures rhétoriques, ou « tropes ». Les discours définis par la rhétorique sont le délibératif (discours tenu pour persuader ou conseiller), le judiciaire (discours tenu pour accuser ou défendre), et l'épidictique (discours tenu pour louer ou blamer). De manière plus générale la rhétorique étudie tout ensemble d'énoncés humains en relation, y compris donc les dialogues quelconques.

Annexe C

La SDRT : un cadre riche pour la formalisation du discours¹

C.1 Introduction

La SDRT est née de la combinaison de deux cadres formels conçus dans les années '80. D'un côté, la DRT (« Discourse Representation Theory ») de Kamp et Reyle [90] a amené de nouvelles perspectives sur la sémantique formelle, en intégrant des concepts de sémantique temporelle dans un formalisme qui permet de tenir compte facilement des portées des quantificateurs, et des références anaphoriques dans des discours contenant plusieurs phrases. D'un autre côté, des chercheurs, notamment en intelligence artificielle, ont proposé des relations rhétoriques permettant de structurer le discours selon les rôles pragmatiques des énoncés qui le constituent, comme dans la RST (« Rhetorical Structure Theory ») de Mann et Thompson [107]. La nouveauté principale proposée par Asher et Lascarides dans la SDRT (créée en 1993), consiste à utiliser des relations rhétoriques pour combler les lacunes de la DRT.

Nous allons d'abord présenter quelques motivations pour construire cette nouvelle théorie, et les moyens que Asher et Lascarides ont choisis pour y parvenir. Cette discussion se base essentiellement sur l'état de la théorie en 1993 [9]. Ce n'est qu'après que nous allons mettre en lumière quelques ajouts et modifications qui ont été faits depuis [11].

C.2 La SDRT en 1993

C.2.1 Position du problème

Un problème classique dans l'interprétation du discours concerne l'interprétation temporelle de ce dernier [90], où on n'arrive pas toujours à déterminer de manière adéquate la structure temporelle des procès décrits dans le discours, comme l'illustrent les exemples suivants :

(1) Jean a ouvert la porte. La chambre était sombre.

¹Ce paragraphe est, en grande partie, inspiré d'un article de Scherrer [155].

(2) Jean a éteint la lumière. La chambre était sombre².

Ces deux discours ont la même structure grammaticale et contiennent des verbes des mêmes types : le premier énoncé réfère à un événement ponctuel dans le passé, tandis que le deuxième énoncé réfère à un état durable dans le passé, dans les deux cas. Si dans (1) l'état recouvre complètement l'événement, ceci n'est pas le cas dans (2) : la pièce ne devient noire qu'à partir du moment où Max éteint la lumière. L'argumentation de Asher est la suivante : si la DRT n'a aucun moyen de distinguer l'interprétation temporelle de (1) et (2), c'est parce qu'elle ne tient pas compte des relations discursives que les deux phrases entretiennent entre elles :

- dans (1), 'la chambre était sombre' correspond à une description de l'*arrière-plan* par rapport à 'Jean a ouvert la porte'.
- dans (2), 'la chambre était sombre' représente la *conséquence* de l'action décrite par 'Jean a éteint la lumière'.

Le but de la SDRT est de se donner les moyens pour inférer ces différences pragmatiques à partir des informations contenues dans les phrases, et des connaissances du monde.

Toutefois, ce n'est pas uniquement au niveau des structures temporelles que la DRT paraît déficiente. Les créateurs de la SDRT montrent que la DRT accepte des discours qui devraient être bloqués pour des raisons pragmatiques :

Jean a eu une merveilleuse soirée hier.
 Il a eu un super dîner.
 Il a mangé du saumon.
 Il a dévoré beaucoup de fromage.
 Il a gagné un concours de danse.
 * Il était d'un joli rose³.

La dernière phrase, dont le pronom 'il' se rapporte au saumon, est agrammaticale dans cette position (elle serait acceptable en quatrième position, juste après 'Il a mangé du saumon'); les phrases précédentes traitant du fromage et du concours de danse ont clos le thème du saumon, de sorte qu'il n'est plus possible d'y référer par le pronom 'il'. Pourtant, les restrictions d'accessibilité de la DRT ne suffisent pas pour expliquer cette référence anaphorique inopportune⁴. De nouveau, la structuration rhétorique du discours apporte une réponse. En s'appuyant sur les travaux de Grosz et Sidner [70], Asher et Lascarides utilisent les relations discursives pour représenter le discours par une structure de graphe. Avec les contraintes de disponibilité⁵ définies sur ces structures, ils parviennent à expliquer pourquoi la dernière phrase de l'exemple précédent est inacceptable.

C.2.2 Les éléments définitoires de la SDRT

Essentiellement, l'architecture de la SDRT est conçue en deux parties principales :

²Ces exemples représentent une traduction des originaux en Anglais, proposés par Asher en [9].

³Cet exemple constitue également une traduction française d'un original de Asher.

⁴En effet, la DRT rend compte des restrictions d'accessibilité imposées par les négations, les quantificateurs et les implications, mais ne tient pas compte des restrictions d'ordre pragmatique.

⁵Ascarides et Asher distinguent entre les contraintes d'accessibilité imposées par la DRT et les contraintes de disponibilité mises en place par la SDRT [11].

- un formalisme de représentation des propositions, hérité de la DRT ;
- un cadre formel – en l’occurrence une logique non-monotone à défauts – pour inférer les relations rhétoriques entre les propositions, et pour les agencer dans une structure (arborescente ou non).

Construire la représentation du discours revient ainsi à (i) déterminer la représentation de chaque proposition, suivant les règles de la DRT⁶ ; (ii) inférer les relations rhétoriques entre les propositions, en s’appuyant sur des informations lexicales, des liens référentiels et des connaissances encyclopédiques.

C.2.3 Les relations rhétoriques principales

Les relations rhétoriques sont formalisées comme des prédicats à deux arguments qui correspondent aux deux structures discursives à relier⁷. Asher [9] propose les relations rhétoriques suivantes :

1. $Narration(\alpha, \beta)$ est conçue comme relation par défaut, pourvu qu’il y ait un topique commun (*Downarrow*) entre les deux phrases. En présence de deux énoncés étiquetés α et β , la SDRT infère qu’ils constituent une séquence narrative. Cette inférence peut toutefois être annulée lorsqu’on dispose d’informations plus précises. Par exemple, $Narration(\alpha, \beta)$ est inférée dans le discours suivant :

Pierre se mit debout ^{α} . Jean le salua ^{β} .

2. $Explanation(\alpha, \beta)$ signifie que β explique α ; elle s’obtient lorsque le texte contient de l’évidence permettant d’inférer un lien de cause à effet entre β et α . Cette évidence peut venir de la présence d’un connecteur comme *parce que*, ou par des informations lexicales et encyclopédiques liées aux énoncés. Ainsi, dans le fameux exemple de Asher, reproduit en traduction ci-dessous, on infère $Explanation$ parce que la base de connaissances contient une règle qui spécifie que ‘ X pousse Y ’ peut être la cause de ‘ Y tombe’ :

Pierre est tombé ^{α} . Jean l’a poussé ^{β} .

3. $Result(\alpha, \beta)$ et équivalent à $Explanation(\beta, \alpha)$. On est obligé d’introduire cette relation à cause d’une contrainte relative à l’ordre textuel des arguments : l’énoncé correspondant au premier argument doit précéder celui du deuxième argument dans le texte :

Jean a éteint la lumière ^{α} . La chambre était sombre ^{β} .

4. $Elaboration(\alpha, \beta)$ est inférée lorsque ce qui est décrit par l’énoncé β fait partie de ce qui est décrit par α . La relation « faire partie de » dépend des définitions des mots contenus dans les énoncés ; elle présuppose donc une organisation arborescente du lexique :

Jean a eu une merveilleuse soirée hier.

Il a eu un super dîner.

⁶Ces représentations, ou structures discursives, sont essentiellement des formes logiques des propositions, contenant aussi des informations temporelles et aspectuelles.

⁷Une structure discursive correspond à une forme logique d’une proposition ou d’un ensemble de propositions.

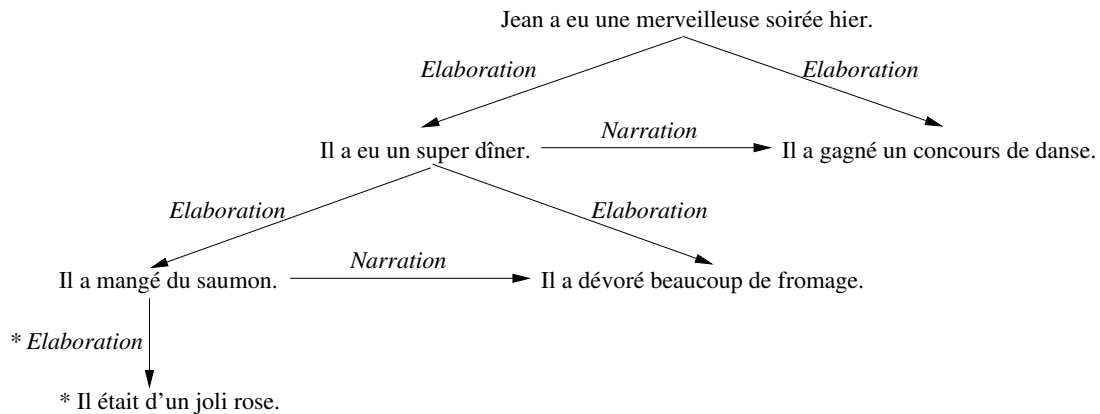


FIG. C.1 – Structure rhétorique d'un discours

5. *Background*(α, β) (*Arrière-plan*) est obtenue quand l'énoncé β décrit les circonstances de ce qui est décrit par α . Par cette définition, β décrit un état, tandis que α décrit un événement :

Jean a ouvert la porte $^{\alpha}$. La chambre était sombre $^{\beta}$.

6. *Parallel*(α, β) indique que les deux énoncés ont une structure et un contenu parallèle, et / ou qu'ils sont reliés par un connecteur de type *aussi* :

Jean aime le sport $^{\alpha}$. Pierre aussi aime le sport $^{\beta}$.

7. *Contrast*(α, β) indique que les deux énoncés ont une structure parallèle mais des contenus contrastants, et / ou qu'ils sont reliés par un connecteur du type *mais* :

Jean aime le sport $^{\alpha}$. Mais il déteste le football $^{\beta}$ ⁸.

Pour l'interprétation des discours, les relations rhétoriques s'appuient en grande partie sur les informations lexicales contenues dans les énoncés. En plus, une base de connaissances encyclopédiques est nécessaire notamment pour inférer les liens de causalité. Ces exigences font à la fois la force et la faiblesse de la SDRT : c'est uniquement en tenant compte d'informations non-syntaxiques qu'on peut s'approcher d'un traitement adéquat des phénomènes sémantiques et pragmatiques, mais la construction de ces ressources à un degré de couverture satisfaisant reste difficile et coûteuse [124].

Les relations rhétoriques servent à générer une structure temporelle des procès du discours. Par exemple, une règle est associée à *Explanation*(α, β) qui précise que le procès décrit par α ne peut précéder temporellement le procès décrit par β . *Background*(α, β) signifie que les deux procès se chevauchent. Ainsi, après avoir inféré les relations rhétoriques pour les exemples (1) et (2) ci-dessus, il en découle également des interprétations temporelles différentes.

L'exemple ci-dessus avec la soirée de Jean est interprété à travers une autre dichotomie au sein des relations discursives : la relation *Narration* est coordonnante, tandis que la relation *Elaboration* est subordonnante, ce qui est montré par les flèches horizontales, respectivement

⁸Cet exemple est emprunté à Asher [9].

verticales, dans la structure de graphe correspondant à cet exemple ; cette structure est montrée dans la figure C.1.

Les restrictions de disponibilité interdisent l'attachement d'une proposition en bas et à gauche du nœud courant. Lorsqu'on se trouve sur le nœud 'Il a gagné un concours de danse', on ne peut plus rien ajouter au nœud 'Il a mangé du saumon', parce que celui-ci se trouve plus bas, à gauche. Cette restriction, la contrainte de frontière droite, bloque l'ajout de la dernière phrase et empêche en même temps la liaison du pronom 'il' à son référent 'saumon'.

C.3 De nouveaux apports : la SDRT en 2003

Depuis 1993, de nombreuses modifications ont été apportées à la SDRT. Ces contributions, pour la plupart publiées dans [11], ont deux visées possibles : d'un côté, elles servent à élargir le domaine d'application de la théorie, en tenant compte de nouveaux phénomènes linguistiques. D'un autre côté, elles servent à simplifier les bases théoriques ou à les rendre plus explicites. Ces deux axes sont complémentaires et largement interdépendants. Souvent, une modification d'un côté nécessite une modification de l'autre. Dans la présentation qui suit, nous résumons quelques-unes de ces modifications apportées à la SDRT.

C.3.1 Extension au dialogue

Un grand domaine de recherche concerne l'intégration des phénomènes du discours dialogique dans la SDRT. Le discours dialogique – par opposition au discours monologique – reflète explicitement une intention d'un locuteur d'interagir avec son (ses) interlocuteur(s). Si l'intention du locuteur d'interagir avec son interlocuteur y est explicitée, cela se fait typiquement par des requêtes ou des questions. Il en découle que l'on doit ajouter des relations rhétoriques pour ce type d'énoncés :

1. $Q\text{-Elab}(\alpha, \beta)$ (Elaboration de question) : β est une question qui élabore α (A et B sont les deux locuteurs) :

A : Je voudrais lire ce livre la semaine prochaine ^{α} .

B : Est-ce bien Jeudi ? ^{β}

2. $P\text{-Elab}(\alpha, \beta)$ (Elaboration de plan)⁹ : β est une requête qui élabore α :

A : Je voudrais prendre ce livre maintenant ^{α} .

B : Allez au deuxième étage, troisième rangée, à gauche ^{β} .

3. $Elab_q(\alpha, \beta)$ (Elaboration_q) : α est une proposition et β est une question dont les réponses possibles constituent une élaboration de α :

A : Je cherche un nouveau roman à lire ^{α} .

B : Sur quoi ? ^{β}

A : Eh ben, un roman néerlandais contemporain.

⁹Dans la SDRT d'origine cette relation était appelée au début *R-Elaboration* (Elaboration de requête).

4. *Result**(α, β) (Résultat*) : l'énonciation de β résulte de l'énonciation de α . Ce n'est donc pas le contenu de β qui résulte de α ; dans ce cas on aurait eu *Result*(α, β) :

A : J'ai froid ^{α} .

B : Ferme la fenêtre alors ! ^{β}

5. *QAP*(α, β) (Paire adjacente question-réponse) : α est une question et β est la réponse correspondante :

A : Quel livre voulez-vous ? ^{α}

B : Je voudrais livre 'X' ^{β} .

6. *IQAP*(α, β) (Paire adjacente indirecte question-réponse) : α est une question et β est une réponse indirecte correspondante :

A : Pourrais-je emprunter ce bouquin ces jours-ci ? ^{α}

B : Ce livre sera disponible dès lundi prochain ^{β} .

L'extension au dialogue nécessite donc la distinction des types de phrases (assertions, questions et requêtes) et l'introduction de relations rhétoriques spécifiques pour ces nouveaux types. Une difficulté potentielle concerne les actes de langage indirects : par exemple, ce qui est une question au niveau locutoire peut avoir une valeur illocutoire de requête (par exemple, 'Peux-tu me passer le sel ?', où une réponse telle que 'Oui' n'est pas satisfaisante). Pour ces cas de figure, Asher et Lascarides [11] introduisent les *dot types* (par exemple, 'question-requête'). Une proposition d'un tel type composé peut figurer dans toutes les relations qui demandent soit une question, soit une requête comme argument.

C.3.2 Les remises en cause en dialogue

Dans le dialogue, il est fréquent qu'un locuteur soit amené à corriger l'autre :

A : Pierre a pris ce bouquin hier.

B : Non, c'est Jean qui l'a pris¹⁰.

Dans cette situation, le contenu d'un énoncé est admis comme étant vrai par le premier interlocuteur, mais est falsifié par la correction apportée par le deuxième locuteur, le contenu de l'énoncé initial étant remplacé par le contenu de la correction. De nouveau, Asher et Lascarides introduisent une relation rhétorique spécifique [11] :

Correction(α, β) : α correspond à ce qui est corrigé et β correspond à la correction apportée. Un axiome précise que cette relation est inférée lorsque α et β sont inconsistants (sinon il ne s'agirait pas d'une correction), et que le focus de α est remplacé par le focus de β .

Cette relation diffère fondamentalement des relations discursives classiques. Etablir une relation de *Narration*, *Elaboration*, etc. entre deux énoncés revient toujours à asserter les deux énoncés. En prononçant le deuxième énoncé, on enchaîne sur le premier, en l'approuvant implicitement. En revanche, la relation *Correction* nie le premier énoncé et n'asserte que le deuxième. Les concepteurs de la SDRT prennent en compte cette différence en distinguant les relations

¹⁰Cet exemple est emprunté à Asher et Lascarides [11].

véridiques (impliquant la vérité de leurs arguments) des relations non-véridiques (ou *divergentes*, n'impliquant pas la vérité de leurs arguments, par exemple *Correction*). Des relations non-véridiques existent aussi dans le monologue, par exemple *Alternation* (Alternative) relie deux propositions connectées par 'ou' ; évidemment, il n'est pas exigé ici que ces propositions soient vraies toutes les deux.

Une autre relation divergente, mais avec une forte composante cognitive (c'est-à-dire, ayant une sémantique spécifiée, au moins partiellement, en termes de croyances et intentions des locuteurs), est la Correction de plan $P\text{-}Corr(\alpha, \beta) : \beta$ implique le fait que son émetteur ne peut pas accepter ou n'est pas capable d'accomplir le but de l'émetteur de α , tel qu'il est précisé en α :

A : Pourrais-je emprunter ce livre lundi prochain ? ^{α}

B : Je suis désolé, mais le livre ne sera disponible que d'ici trois semaines ^{β} .

C.3.3 Formalisation de la cohérence discursive

Les relations *Parallel* (Parallèle) et *Contrast* (Contraste) sont scalaires (non-binaires), dans le sens où les contrastes et parallèles peuvent être plus ou moins marqués selon les énoncés en question. Par exemple, l'exemple ‡ présente un contraste plus marqué que † :

(†) Jean aime collectionner des voitures classiques. Mais sa voiture préférée est un Ford de 1999.

(‡) Jean aime collectionner des voitures classiques. Mais il déteste le football.

Il est utile de connaître le degré de parallèle ou de contraste dans les cas où le raisonnement logique ne parvient pas à sélectionner une seule structure du discours, et que différentes interprétations incompatibles restent en concurrence. Asher envisage ainsi de favoriser l'interprétation à valeurs maximales [9] : la contrainte de maximisation pour les relations de *Parallel* et *Contrast* exige la maximisation de la force de l'attribution de la polarité pour ainsi choisir la relation de *Contrast* ou *Parallel* qui est la plus forte.

Par la suite, cette contrainte de maximisation a été étendue à d'autres phénomènes, devenant le Principe de maximisation de la cohérence discursive (« Maximize Discourse Coherence Principle » – MDC) : la cohérence d'un discours ne dépend désormais pas seulement de la valeur des relations scalaires (dont fait dorénavant partie la relation de *Narration*), mais également du nombre des relations et connexions anaphoriques obtenues [11].

Ainsi, selon Asher et Lascarides, le discours est interprété ainsi que sa cohérence soit maximale, ce qui revient à trois principes :

1. plus le nombre de connexions rhétoriques (via des relations de discours) entre deux énoncés ou constituants discursifs est grand, plus l'interprétation du discours composé de ce deux énoncés ou constituants discursifs est cohérence ;
2. plus le nombre d'expressions anaphoriques à l'antécédent résolu est grand, plus l'interprétation du discours est cohérente ;
3. pour les relations rhétoriques scalaires (par exemple, *Narration* ou *Contrast*), dont la qualité dépend des morceaux de texte qu'elles connectent (par exemple, la qualité de

Contraste dépend de la mesure où les sémantiques des morceaux connectés sont différents), une interprétation qui maximise la qualité de ces relations est plus cohérente qu'une autre qui ne le fait pas.

Ce principe MDC introduit une composante quantitative de plus en plus importante dans la SDRT, ce qui ouvre la voie vers des modélisations et implémentations computationnelles de la théorie.

C.3.4 La sous-spécification

Typiquement, les phrases contenant des quantificateurs peuvent être ambiguës. Deux approches sont concevables pour représenter cette ambiguïté formellement. On peut choisir de générer, dans la composante sémantique, l'ensemble des formes logiques liées aux différentes portées des quantificateurs et de passer cet ensemble à la composante pragmatique qui agit comme un filtre, en choisissant la bonne interprétation à l'aide du contexte et écartant les mauvaises. Ou bien on choisit de générer une seule forme logique pour les différentes interprétations, une forme qui représente en elle-même cette ambiguïté. La composante pragmatique doit alors « remplir les trous » contenus dans cette structure incomplète. Cette dernière approche illustre le fonctionnement de la sous-spécification.

L'introduction des formes sous-spécifiées dans la SDRT est transversale ; la sous-spécification s'utilise tant au niveau de la construction des formes logiques représentant les propositions, comme évoqué ci-dessus, qu'au niveau de de l'inférence des relations rhétoriques entre les propositions. A chaque fois qu'on ajoute une proposition à la représentation globale du discours (cf. le schéma dans la figure C.1), on procède à une mise à jour de cette représentation. Une partie de cette mise à jour concerne l'inférence des relations rhétoriques qui attachent cette nouvelle proposition au reste du discours précédemment analysé. Dans la SDRT actuelle, on utilise la sous-spécification pour déclencher ce raisonnement logique. On fait l'hypothèse que cette nouvelle proposition s'attache par au moins une relation rhétorique à au moins une proposition déjà analysée ; cette hypothèse est représentée par le prédicat sous-spécifié $?_{\rho}(?_{\alpha}, \beta)$ – la nouvelle proposition β est reliée par une relation encore inconnue $?_{\rho}$ à un élément de discours encore inconnu $?_{\alpha}$. A l'aide des axiomes définissant les relations rhétoriques et des informations lexicales et encyclopédiques, le moteur d'inférence résout ces sous-spécifications graduellement par unification. Ce mécanisme remplace la fonction de « mise à jour », de la version de 1993 de la SDRT. La mise à jour passe d'un mécanisme procédural à un mécanisme purement déclaratif, ce qui rend le processus plus intuitif. Cependant, la sous-spécification, avec l'introduction des variables d'ordre supérieur, nécessite un système logique plus puissant.

C.3.5 En guise de conclusion : une théorie modulaire

La version de 1993 contient deux types d'axiomes : le premier type sert à définir les conditions nécessaires pour obtenir une relation rhétorique particulière, comme la condition du lien causal entre deux énoncés reliés par *Explanation*. Le deuxième type sert à inférer la structure temporelle décrite par le discours une fois les relations rhétoriques obtenues, comme le fait que l'événement ou procès de α ne peut précéder celui de β dans *Explanation*(α, β). Du point de vue

formel, les deux types d'axiomes ne se distinguent pas ; c'est uniquement le caractère nécessaire – présent dans les derniers, mais absent dans les premiers axiomes – qui fait la différence.

Dans la version de 2003, les concepteurs de la SDRT reprennent la base théorique et l'articulent autour de différents modules reliés aux différentes tâches. Les deux types d'axiomes se retrouvent alors dans deux modules différents. Asher et Lascarides avancent une motivation pratique pour la modularisation de la théorie. Selon eux, il faut séparer la construction des représentations discursives de leur interprétation, car il existe des énoncés dont on n'est pas en mesure de déterminer la valeur de vérité (interprétation), mais dont on peut tout de même déterminer le rôle rhétorique dans le discours (construction).

Avec la modularisation, l'architecture globale de la SDRT s'est considérablement compliquée, mais en même temps clarifiée. Chaque module est géré par une logique différente adaptée à la tâche correspondante. Sans entrer dans les détails, nous mentionnons ci-dessous les modules principaux :

- une logique pour représenter les connaissances du monde (encyclopédiques) ; c'est avant tout une base de données décrivant des relations causales ;
- une logique pour la description du contenu ; ce système sert à construire les représentations sémantiques des propositions ; ces représentations peuvent être sous-spécifiées (quantifications) ;
- une logique pour inférer les relations rhétoriques ; elle utilise les représentations sémantiques sous-spécifiées, les connaissances du monde et le lexique ; c'est le système principal de la théorie qui « colle ensemble » les différentes représentations – d'où le nom, en Anglais, de *glue logic* ; c'est ici que les conditions pour inférer les relations rhétoriques sont définies ;
- une logique pour interpréter la représentation discursive ; cela a pour but d'assigner des valeurs de vérité aux propositions connectées par les relations rhétoriques, et ensuite inférer les conséquences temporelles des relations rhétoriques ; c'est ici que les axiomes du deuxième type (cf. ci-dessus) sont localisés.

Les modifications apportées à la SDRT ces dernières années sont nombreuses. Nous avons présenté, de manière concise et plutôt informelle, les idées les plus originales et importantes pour les recherches décrites dans ce document. Quelques idées ont pour but l'extension de la théorie à de nouveaux phénomènes linguistiques, comme les relations introduites pour gérer les caractéristiques du dialogue. D'autres concernent les bases théoriques, comme la modularisation.

Bibliographie

- [1] Adam C., Gaudou B., Herzig A., Longin D., « OCC's Emotions : a Formalization in BDI Logic », *Proceedings of AIMS*, Springer, 2006.
- [2] Allwood J., *Linguistic Communication as Action and Cooperation*, Gothenburg Monographs in Linguistics 2, 1976.
- [3] Allwood J., « Obligations and Options in Dialogue », *Think Quarterly* 3(1) : 12-34, 1994.
- [4] Amgoud L., Maudet N., Parson S., « Modelling Dialogue Using Argumentation », *Proceedings of the 4th International Conference on Multi-Agent Systems ICMAS'00*, 2000.
- [5] Anscombe J. -C., « Marqueurs et hypermarqueurs de dérivation illocutoire : notion et problèmes », *Cahiers de linguistique française* 3 : 75-124, 1981.
- [6] Anscombe J. -C., « Topique or not topique : formes topiques intrinsèques et formes topiques extrinsèques », *Journal of Pragmatics* 24, 1995.
- [7] Anscombe J. -C., Ducrot O., « L'argumentation dans la langue », *Langages* 10(42) : 5-27, 1976.
- [8] Appelt D., *Planning English Sentences*, Cambridge University Press, 1985.
- [9] Asher N., *Reference to Abstract Objects in Discourse*, Kluwer, Dordrecht, 1993.
- [10] Asher N., Lascarides A., « Questions in Dialogue », *Linguistics and Philosophy* 21(3) : 237-309, 1998.
- [11] Asher N., Lascarides A., *Logics of Conversation*, Cambridge University Press, 2003.
- [12] Asher N., Gillies A., « Common Ground, Corrections, and Coordination », *Argumentation* 17 : 481-512, 2003.
- [13] Asher N., « Discourse Topic », *Theoretical Linguistics* 30 : 161-201, 2004.
- [14] Asher N., Prévot L., Vieu L., « Setting the Background in Discourse », *Discours* 1, 2007 (<http://www.discours.revues.org>).
- [15] Asher N., Reese B., « Prosody and the Interpretation of Tag Questions », in E. Puig-Waldmüller (Ed.), *Proceedings of Sub 11*, Pompeu-Fabra University, Barcelona, 2007.
- [16] Austin J. L., *Quand dire, c'est faire*, Seuil, 1999.
- [17] Bakhtine M., *Esthétique de la création verbale*, Gallimard, 1984.
- [18] Benotti L., « Incomplete Knowledge and Tacit Action : Enlightened Update in a Dialogue Game », *Proceedings of the Tenth Workshop on Semantics and Pragmatics of Dialogue SEMDIAL – DECALOG*, Trento, 2007.

- [19] Berger A., *La communication entre agents de communauté mixtes : un langage de conversation expressif pour agents artificiels*, Thèse de doctorat, Institut National Polytechnique de Grenoble, 2006.
- [20] Bilange E., *Dialogue personne-machine*, Hermès Science, 1992.
- [21] Blackburn P., Bos J., *Representation and Inference for Natural Language : A First Course in Computational Semantics*, CSLI Publications, 2001.
- [22] Blackburn P., Bos J., *Working with Discourse Representation Theory : An Advanced Course in Computational Semantics*, CSLI Publications, 2001.
- [23] Bod, R., *Beyond Grammar : An Experience Based Theory of Language*, CSLI Publications, 1998.
- [24] Brandom R., *Articulating Reasons : An Introduction to Inferentialism*, Harvard University Press, Cambridge, Massachussets, 2000.
- [25] Branigan H., « Perspectives on Multi-party Dialogue », *Research on Language and Computation* 4 : 153-177, 2006.
- [26] Brassac C., « Speech Act and Conversation Sequencing », *Pragmatics and Cognition* 2(1) : 191-205, 1994.
- [27] Bratman M. E., *Intention, Plans and Practical Reason*, Harvard University Press, 1987.
- [28] Brown P., Levinson S., *Politeness : Some Universals in Language Use*, Cambridge University Press, 1987.
- [29] Bruxelles S., Ducrot O., Raccach P. -Y., « Argumentation and the Lexical Topical Fields », *Journal of Pragmatics* 24, 1995.
- [30] Bunt H., « Dynamic Interpretation and Dialogue Theory », in *The Structure of Multimodal Dialogue*, John Benjamins, 1996.
- [31] Bunt H., « Dialogue Pragmatics and Context Specification », H. Bunt, W. Black (Eds.) *Abduction, Belief and Context in Dialogue : Studies in Computational Pragmatics*, John Benjamins, 2000.
- [32] Bunt H., « The Missing Link », *Lecture Notes in Computer Science* 4629, 2007.
- [33] Bunt H., Black W. (Eds.), *Abduction, Belief and Context in Dialogue : Studies in Computational Pragmatics*, John Benjamins, 2000.
- [34] Caelen J., « Strategies of Dialogue », in C. Burileanu (Ed.), *Speech Technology and Human-Computer Dialogue*, Romanian Academy Press, 2003.
- [35] Caelen J., Xuereb A., *Interaction et pragmatique - jeux de dialogue et de langage*, Hermès Science, 2007.
- [36] Caron J., « Les opérateurs discursifs comme instructions de traitement », *Verbum* 7 : 149-164, 1994.
- [37] Cavazza M., Charles F., Mead S. J. « Planning Characters' Behaviour in Interactive Storytelling », *Proceedings of JVCA 2002*, 2002.
- [38] Clark H. H., *Arenas of Language Use*, CSLI Publications, 1993.

- [39] Clark H. H., *Using Language*, Cambridge University Press, 1996.
- [40] Clark H. H., Carlson T. B., « Hearers and Speech Acts », *Language* 58 : 332-373, 1982.
- [41] Clark H. H., Schaefer E., « Contributing to Discourse », *Cognitive Science* 13(2) : 259-294, 1989.
- [42] Chambreuil M. (Ed.), *Sémantiques*, Hermès Science, 1998.
- [43] Chu-Carroll J., Carberry S., « Collaborative Response Generation in Planning Dialogues », *Computational Linguistics* 24(3), 1998.
- [44] Cohen P. R., Perrault C. R., « Elements of a Plan Based Theory of Speech Acts », *Cognitive Science* 3 : 177-212, 1979.
- [45] Cohen P. R., Levesque H. J. « Intention is Choice with Commitment », *Artificial Intelligence* 42 : 213-261, 1990.
- [46] Colineau N., *Etude des marqueurs discursifs dans le dialogue finalisé*, Thèse de doctorat, Université Joseph Fourier, Grenoble, 1997.
- [47] Corblin F., « Presuppositions and Commitment Stores », *Proceedings of the 7th Workshop on the Semantics and Pragmatics of Dialogue SEMDIAL – DIABRUCK*, Wallerfangen, 2003.
- [48] Cristea D., Ide N., Romary L., « Veins Theory. An Approach to Global Cohesion and Coherence », *Proceedings of COLING/ACL 1998*, Montreal, 1998.
- [49] Culioli A., *Pour une linguistique de l'énonciation*, Tomes 1, 2, 3, Ophyrus, 1990-1993.
- [50] Danlos L., El-Ghali D., « A complete integrated NLG system using AI and NLU tools », *Proceedings of COLING 2002*, Taipei, 2002.
- [51] Danlos L., Gaiffe B., Roussarie L., « Document Structuring à la SDRT », *Proceedings of the International Workshop on Generation, ACL*, Toulouse, 2003.
- [52] Demazeau Y., Muller J. -P., *De-centralized AI*, North-Holland, 1990.
- [53] Deransart P., Małuzýnski J., *A Grammatical View of Logic Programming*, MIT Press, 1993.
- [54] Dubois J., Giacomo M., Guespin L., Marcellesi C., Marcellesi J.-B., Mevel J.-P., *Dictionnaire de linguistique*, Larousse-Bordas, 2001.
- [55] Ducrot O., *Les mots du discours*, Editions de Minuit, 1980.
- [56] Ducrot O., *Les échelles argumentatives*, Editions de Minuit, 1980.
- [57] Ducrot O., *Le dire et le dit*, Editions de Minuit, 1987.
- [58] Ducrot O., « Les modificateurs déréalisants », *Journal of Pragmatics* 24, 1995.
- [59] Egg M., Redeker G., « Underspecified Discourse Representation », *Constraints in Discourse*, John Benjamins, 2007.
- [60] Elhadad M., « Using argumentation in text generation », *Journal of Pragmatics* 24 : 189-220, 1995.
- [61] Fillmore Ch., « The Case for Case », in E. Bach and R. Harms (Eds.), *Universals in Linguistic Theory*, Holt, Rinehart and Winston, New York, 1968.

- [62] Frege G., « Sens et dénotation », in G. Frege (Ed.), *Ecrits logiques et philosophiques*, Seuil, 1971.
- [63] Gallier J., *Logic for Computer Science*, John Wiley and Sons, 1986.
- [64] Ginzburg J., Fernandez R. « Scaling up from Dialogue to Multilogue : Some Principles and Benchmarks », *Proceedings of ACL 2005*, 2005.
- [65] Ginzburg J., *Semantics and Interaction in Dialogue*, CSLI Publications, forthcoming.
- [66] Gardent C, Kow E., « A Symbolic Approach to Near-Deterministic Surface Realisation using Tree Adjoining Grammar », *Proceedings of ACL*, Prague, 2007.
- [67] Grice H. P., « Meaning », *Philosophical Review* 66 : 377-388, 1957.
- [68] Grice H. P., « Logic and Conversation », in P. Cole and J. L. Morgan (Eds.), *Syntax and Semantics 3* :41-58, Academic Press, 1975.
- [69] Grice H. P., *Studies in the Way of Words*, Harvard University Press, Cambridge, Massachusetts, 1989.
- [70] Grosz B. J., Sidner C. L., « Attention, Intentions, and the Structure of Discourse », *Computational Linguistics* 12(3) : 175-204, 1986.
- [71] Gupta S., Walker M. A., Romano D. M., « Generating Politeness in Task Based Interaction : An Evaluation of the Effect of Linguistic Form and Culture », *Proceedings of the 11th European Workshop on Natural Language Generation ENLG'07*, Schloss Dagstuhl, 2007.
- [72] Hagiescu, I., *Création automatique des ontologies de domaine pour les systèmes de dialogue homme-machine*, Mémoire pour le Diplôme d'Ingénieur, Université « Politehnica » de Bucarest (Encadrants : C. Burileanu et V. Popescu), 2007.
- [73] Hamblin C. L., *Fallacies*, Methuen and Co. Ltd., London, 1970.
- [74] Hamblin C. L., « Mathematical Models of Dialogue », *Theoria* 37 : 130-155, 1971.
- [75] Heidegger M., *Acheminement vers la parole*, Gallimard, 2003.
- [76] Hobbs J. R., « Coherence and Coreference », *Cognitive Science* : 27-90, 1979.
- [77] Hobbs J. R., *Literature and Cognition*, CSLI Publications, 1990.
- [78] Hobbs J. R., Stickel M., Appelt D., Martin P., « Interpretation as Abduction », *Artificial Intelligence* 63 : 69-142, 1993.
- [79] Hovy E., « Pragmatics and Natural Language Generation », *Artificial Intelligence* 43 : 153-197, 1990.
- [80] Hovy E., « Automated discourse generation using discourse structure relations », *Artificial Intelligence* 65 : 341-386, 1993.
- [81] Imberdis L., Caelen J., « Génération d'actes illocutoires pour le dialogue », *Actes de l'Atelier Génération automatique des textes*, Grenoble, 1997.
- [82] Irmer M., « Bridges Between Events. Indirect Reference to Eventualities », *Proceedings of the 3rd ACL SIGGen Workshop Constraints in Discourse*, Potsdam, 2008.

- [83] Isakowits T., « Can We Transform Logic Programs into Attribute Grammars ? », Technical Report, New York University, 1991.
- [84] Iten C., « The relevance of Argumentation Theory », *Lingua* 110(9) : 665-699, 2000.
- [85] Jayez J., Dagnat M., « The Interpretation of Continuative Cues in SDRT », *Proceedings of the 3rd ACL SIGGen Workshop Constraints in Discourse*, Potsdam, 2008.
- [86] Jokinen K., *Response Planning in Information-Seeking Dialogues*, Ph D Thesis, University of Manchester, 1994.
- [87] Jokinen K., « Ellipsis Generation in Communicative Dialogues », *Proceedings of JANLP'95*, 1995.
- [88] Jokinen K., « Adequacy and Evaluation », *Proceedings of the ECAI'96 Workshop Gaps and Bridges : New Directions in Planning and Natural Language Generation*, 1996.
- [89] Jurafsky D., « Pragmatics and Computational Linguistics », in R. Horn and G. Ward (Eds.), *Handbook of Pragmatics*, Blackwell, Oxford, 2001.
- [90] Kamp H., Reyle U., *From Discourse to Lexicon : Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, Kluwer Academic Publishers, 1993.
- [91] Kibble R., « Reasoning About Propositional Commitments in Dialogue », *Research on Language and Computation* 4 : 179-202, 2006.
- [92] Kibble R., « Generating Coherence Relations via Internal Argumentation », *Journal of Logic, Language and Information* 16 : 387-402, 2007.
- [93] Kibble R., Piwek P., « Introducing Dialogue Games », *Lecture Notes, European Summer School on Logic, Language and Information*, Dublin, 2007.
- [94] Klabbers E., *Segmental and Prosodic Improvements to Speech Generation*, Ph D Thesis, University of Eindhoven, 2000.
- [95] Kruijff-Korbayová, I., *The Dynamic Potential of Topic and Focus : A Praguian Approach to Discourse Representation Theory*, Ph D Thesis, Charles University, Prague, 1998.
- [96] Kruijff-Korbayová, I., Ericsson, S., García C., Jonson R., Karagjosova E., Manchón P., Rodríguez K. J., Quesada J. F., *Improving System Output Using Information State*, Deliverable D5.1, SIRIDUS Project, 2002 (<http://www.ling.gu.se/projekt/siridus>).
- [97] Landragin F., *Dialogue personne-machine multimodal*, Hermès Science, 2005.
- [98] Larsson S., Traum D., « Information State and Dialogue Management in the TRINDI Dialogue Move Engine Toolkit », *Journal of Natural Language Engineering* 1(1), 2000.
- [99] Lascarides A., Oberlander J., « Abducing Temporal Discourse », *Aspects of Automated Natural Language Generation*, Springer, 1992.
- [100] Lascarides A., Asher N., « Segmented Discourse Representation Theory : Dynamic Semantics with Discourse Structure », H. Bunt, R. Muskens (Eds.), *Computing Meaning* 3, Kluwer Academic Publishers, Dordrecht, 2004.
- [101] Lascarides A., Asher N., « Grounding and Correcting Commitments in Dialogue », sous presse (<http://homepages.inf.ed.ac.uk/alex/papers.html>).

- [102] Levelt W. J. M., *Speaking : From Intention to Articulation*, MIT Press, 1989.
- [103] Levinson S. C., « Activity Type and Language », *Linguistics* 17 : 365-399, 1979.
- [104] Levinson S. C., *Pragmatics*, Cambridge University Press, 1983.
- [105] Lund S. N., « Dynamisme et topoï argumentatifs », *Journal of Pragmatics* 30 : 615-626, 1998.
- [106] Luzatti D., *Le dialogue verbal homme-machine*, Masson, 1995.
- [107] Mann W. C., Thompson S. A., « Rhetorical Structure Theory : Toward a Functional Theory of Text Organization », *Text* 8(3) : 243-281, 1988.
- [108] Marcu D., *The Rhetorical Parsing, Summarization and Generation of Natural Language Texts*, Ph D Thesis, University of Toronto, 1997.
- [109] Marcu D., *The Theory and Practice of Discourse Parsing and Summarization*, MIT Press, Boston, 2000.
- [110] Maudet N., Chaïb-Draa, B., « Commitment-based and Dialogue Game-based Protocols – New Trends in Agent Communication Language », *Knowledge Engineering* 17(2) : 157-179, 2002.
- [111] Maudet N., Muller P., Prévot L., « Tableaux conversationnels en SDRT », *Actes de Traitement Automatique des Langues Naturelles*, Fès, 2004.
- [112] Maudet N., Muller P., Prévot L., « Social Constraints on Rhetorical Relations in Dialogue », *Proceedings of the 2nd ACL SIGGen Workshop Constraints in Discourse*, Maynooth, 2006.
- [113] McKeown K., « Discourse strategies for generating natural-language text », *Artificial Intelligence* 27 : 1-42, 1985.
- [114] McKeown K., Pan S., « Prosody modelling in concept-to-speech generation : methodological issues », *Philosophical Transaction of the Royal Society* 358 : 1419-1431, 2000.
- [115] McShane M., Nirenburg S., « OntoSem Methods for Processing Semantic Ellipsis », *Proceedings of HLT/NAACL Workshop on Computational Lexical Semantics*, USA, 2004.
- [116] McTear M. F., « Spoken Language Technology : Enabling the Conversational User Interface », *ACM Computing Surveys* 34(1), 2002.
- [117] Minker W., Bennacef S., *Parole et dialogue homme-machine*, CNRS Editions, 2001.
- [118] Moeschler J., *Modélisation du dialogue - représentation de l'inférence argumentative*, Hermès Science, 1989.
- [119] Moeschler J., Reboul A., *Dictionnaire encyclopédique de pragmatique*, Seuil, 1994.
- [120] Muller P., Prévot L., « Conversation sous les topiques, du contenu propositionnel à la structure du dialogue », *Information – Interaction – Intelligence*, hors série : 179-196, 2002.
- [121] Neumann, G., « Interleaving Natural Language Parsing and Generation Through Uniform Processing », *Journal of Artificial Intelligence* 99 : 121-163, 1998.
- [122] Nguyen H., *Dialogue homme-machine et modélisation de multi-session*, Thèse de doctorat, Université Joseph Fourier, Grenoble, 2005.

- [123] Nguyen H., Caelen J., « Multi-session Management in Human-Computer Dialogue », *Proceedings of IBERAMIA 2004*, 2004.
- [124] Nirenburg S., Raskin V., *Ontological Semantics*, MIT Press, 2004.
- [125] Nogier J. -F., *Génération automatique de langage et graphes conceptuels*, Hermès Science, 1991.
- [126] Oberlander J., Lascarides A., « Laconic Discourses and total eclipses : abduction in DICE », *Abduction, Beliefs and Context : Studies in Computational Pragmatics*, John Benjamins, 2000.
- [127] Oh A., Rudnicky A., « Stochastic Natural Language Generation for Spoken Dialog Systems », *Computer Speech and Language* 16, 2002.
- [128] Pereira F., Shieber S., *Prolog for Natural-Language Analysis*, CSLI Publications, 1987.
- [129] Perrault C. R., Allen J. F., « A Plan Based Analysis of Indirect Speech Acts », *American Journal of Computational Linguistics* 6(3) : 167-182, 1980.
- [130] Piwek P., *Logic, Information and Conversation*, Ph D Thesis, University of Eindhoven, 1998.
- [131] Piwek P., van Deemter K., *Constraint-based Natural Language Generation : A Survey*, Technical Report no. 2006/03, Department of Computing, The Open University, UK, 2006.
- [132] Poesio M., Traum D., « The TRAINS Project : A Case Study in Building a Conversational Planning Agent », *Journal of Experimental and Theoretical AI*, 7(4) : 7-48, 1995.
- [133] Poesio M., Traum D., « Conversational Actions and Discourse Situations », *Computational Intelligence* 13(3) : 309-347, 1997.
- [134] Pollack M. E., « Plans as Complex Mental Attitudes », in P. R. Cohen, J. Morgan and M. E. Pollack (Eds.), *Intentions in Communications*, MIT Press, Cambridge, Massachusetts, 1990.
- [135] Ponton C., *Formalisme et architecture d'un système de génération du langage naturel ouvert multi-applications*, Thèse de doctorat, Université Stendhal, Grenoble, 1996.
- [136] Popescu V., Caelen J., Burileanu C., « Logic-Based Rhetorical Structuring for Natural Language Generation in Human-Computer Dialogue », *Lecture Notes in Computer Science* 4629, 2007.
- [137] Popescu V., Caelen J., Burileanu C., « Using Speech Acts in Logic-Based Rhetorical Structuring for Natural Language Generation in Human-Computer Dialogue », *Proceedings of the 8th ACL Workshop on Discourse and Dialogue SIGDial 2007*, Antwerp, 2008.
- [138] Popescu V., Caelen J., Burileanu C., « Pragmatically-Motivated Utterance Fine-Tuning in Natural Language Generation for Human-Computer Multi-Party Dialogue », *Research on Language and Computation*, 2008 (accepted with major revisions).
- [139] Popescu V., Caelen J., Burileanu C., « Contrôle rhétorique de l'ellipse sémantique en génération du langage pour le dialogue homme-machine à plusieurs locuteurs », *Traitement Automatique des Langues* 49(1), 2008.

- [140] Popescu V., Caelen J., « Contextual Filtering of Rhetorical Relations in Discourse Structuring for Natural Language Generation in Human-Computer Dialogue », *Proceedings of the 3rd ACL SIGGen Workshop Constraints in Discourse*, Potsdam, 2008.
- [141] Popescu V., Caelen J., « Contrôle rhétorique de la génération des connecteurs concessifs en dialogue homme-machine », *Actes de Traitement Automatique des Langues Naturelles*, Avignon, 2008.
- [142] Popescu-Belis A., Zufferey S., « Contrasting the Automatic Identification of Two Discourse Markers in Multi-Party Dialogues », *Proceedings of the 8th ACL Workshop on Discourse and Dialogue SIGDial 2007*, Antwerp, 2007.
- [143] Porayska-Pomsta K., *Influence of Situational Context on Language Production : Modelling Teachers' Corrective Responses*, Ph D Thesis, University of Edinburgh, 2003.
- [144] Power R., « The Organisation of Purposeful Dialogues », *Linguistics* 17 : 107-152, 1979.
- [145] Raccach P. -Y., « Modelling Argumentation and Modelling with Argumentation », *Argumentation* 4 : 447-483, 1990.
- [146] Ratnaparkhi A., « Trainable Approaches to Surface Natural Language Generation and their Application to Conversational Dialog Systems », *Computer Speech and Language* 16, 2002.
- [147] Reiter E., Dale R., *Building Natural Language Generation Systems*, Cambridge University Press, 2000.
- [148] Roulet E., Auchlin A., Moeschler J., Rubattel C., Schelling M., *L'articulation du français dans le discours contemporain*, Peter Lang, Berne, 1985.
- [149] Roussarie L., *Un modèle théorique d'inférence de structures sémantiques et discursives dans le cadre de la génération automatique de textes*, Thèse de doctorat, Université Paris 7, 2000.
- [150] Russell S., Norvig P., *Artificial Intelligence : A Modern Approach*, Prentice Hall, 2003.
- [151] Sabah G., « A Model for Interaction Between Cognitive Processes », *Proceedings of COLING'90* 3 : 446-448, Helsinki, 1990.
- [152] Sadek D., Bretier P., Panaget F., « ARTIMIS : Natural Dialogue Meets Rational Agency », *Proceedings of the 5th International Joint Conference on Artificial Intelligence IJCAI'95*, Yokohama, 1995.
- [153] Schlangen D., Lascarides A., Copestake A., « Resolving Underspecification Using Discourse Information », *Proceedings of Bi-Dialog*, Bielefeld, 2001.
- [154] Schlangen D., *A Coherence-based Approach to the Interpretation of Non-Sentential Utterances in Dialogue*, Ph D Thesis, University of Edinburgh, 2003.
- [155] Scherrer Y., « Nouveaux développements de la SDRT », *Temps et interprétation, Travaux neuchâtelois de linguistique* 45, Université de Neuchâtel, 2006.
- [156] Searle J. R., *Speech Acts : An Essay in Philosophy of Language*, Cambridge University Press, 1969.

- [157] Searle J. R., Vanderveken D., *Foundations of Illocutionary Logic*, Cambridge University Press, 1985.
- [158] Sgall, P., Hajičová E., Panevová J., *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*, Reidel, Dordrecht, 1986.
- [159] Sperber D., Wilson D., *Relevance*, Harvard University Press, Cambridge, Massachusetts, 1989.
- [160] Sporelander C., Lascarides A., « Using automatically labelled examples to classify rhetorical relations : an assessment », *Journal of Natural Language Engineering* 14(3) : 369-416, 2006.
- [161] Staudacher M., *SDRT Reformulated using DPL*, Term Paper, Bielefeld University, 2005.
- [162] Staudacher M., « A Problem with SDRT's Formalization », *Proceedings of the 3rd ACL SIGGen Workshop Constraints in Discourse*, Potsdam, 2008.
- [163] Stent A., « Rhetorical Structure in Dialog », *Proceedings of the International Conference on Natural Language Generation INLG'00*, 2000.
- [164] Stent A., *Dialogue Systems as Conversational Partners : Applying Conversational Acts Theory to Natural Language Generation for Task-Oriented Mixed-Initiative Spoken Dialogue*, Ph D Thesis, University of Rochester, 2001.
- [165] Stent A. « A Conversation Acts Model for Generating Spoken Dialogue Contributions », *Computer Speech and Language* 16, 2002.
- [166] Stone M., *Modality in Dialogue : Planning, Pragmatics and Computation*, Ph D Thesis, University of Pennsylvania, 1998.
- [167] Stone M., Doran C., Webber B., Palmer M., « Microplanning with Communicative Intentions : The SPUD System », *Computational Intelligence* 19 : 311-381, 2003.
- [168] Striegnitz K., *Generating Anaphoric expressions : Contextual Reasoning in Sentence Planning*, Ph D Thesis, University of Saalanes & Université de Nancy, 2005.
- [169] Theune M., *From Data to Speech : Language Generation in Context*, Ph D Thesis, University of Eindhoven, 2000.
- [170] Theune M., *Natural Language Generation for Dialogue : system survey*, Parlevink Language Engineering Group, University of Twente, 2003.
- [171] Thue D., Bulitko V., Spetch M., Wasylishen E., « Interactive Storytelling : A Player Modelling Approach », *Proceedings of AIIDE 2007*, 2007.
- [172] Traum D., *A Computational Theory of Grounding in Natural Language Conversation*, Ph D Thesis, University of Rochester, 1994.
- [173] Traum D., « A Reactive-Deliberative Model of Dialogue Agency », in J. -P. Müller, M. Wooldridge and N. Jennings (Eds.), *Intelligent Agents III, Lecture Notes in Artificial Intelligence* 1193 : 157-171, 1997.
- [174] Traum D., « Speech Acts for Dialogue Agents », in M. Wooldridge, A. Rao (Eds.) *Foundations of Rational Agents*, Kluwer, 1999.

- [175] Traum D., « Issues in Multi-party Dialogues », in F. Dignum (Ed.), *Advances in Agent Communication, Lecture Notes in Artificial Intelligence 2922* : 201-221, 2004.
- [176] Traum D., Fleischman M., Hovy E., « NL Generation for Virtual Humans in a Complex Social Environment », *Proceedings of the AAAI 2003 Spring Symposium on Natural Language Generation in Spoken and Written Dialogue*, Palo Alto, 2003.
- [177] Traum D., Hinkelman E., « Conversation Acts in Task-Oriented Spoken Dialogue », *Computational Intelligence* 8(3), 1992.
- [178] Traum D., Poesio M., « Conversational Actions and Discourse Situations », *Computational Intelligence* 13(3) : 309-347, 1997.
- [179] TRINDI Consortium, *The TRINDI Book*, 2001 (<http://www.ling.gu.se/projekt/trindi>).
- [180] Trognon A., Brassac C., « L'enchaînement conversationnel », *Cahiers de linguistique française* 13 : 76-107, 1992.
- [181] Trudgill P., *Sociolinguistics : an Introduction to Language and Society*, Fourth Edition, Penguin Books, 2000.
- [182] Tsang E., *Foundations of Constraint Satisfaction*, Academic Press, 1993.
- [183] Tufiş D., Cristea D., Stamou S., « BalkaNet : Aims, Methods, Results and Perspectives. A General Overview », *Romanian Journal of Information Science and Technology* 7(1-2) : 9-43, 2004.
- [184] van Deemter K., Kibble R. (Eds.), *Information Sharing : Reference and Presupposition in Language Generation and Interpretation*, CSLI Publications, 2004.
- [185] Vanderveken D., *Meaning and Speech Acts*, Cambridge University Press, 1990-1991.
- [186] Veale T., « Qualia Extraction and Creative Metaphor in WordNet », *Proceedings of IJ-CAI'03 – Workshop on Creative Systems*, Acapulco, 2003.
- [187] Vernant D., « Modèle projectif et structure actionnelle du dialogue informatif », *Du dialogue, Recherches sur la philosophie du langage* 14 : 295-314, Vrin, 1992.
- [188] Vernant D., *Du discours à l'action*, Presses Universitaires de France, 1997.
- [189] Vion R., *La communication verbale. Analyse des interactions*, Hachette Supérieur, 1992.
- [190] Vygotski L. S., *Conscience, inconscient, émotions*, La Dispute, 2003.
- [191] Wilson D., Sperber D., « L'interprétation des énoncés », *Communications* 22 : 80-94, 1979.
- [192] Wittgenstein L., *Philosophical Investigations*, McMillan, Oxford University Press, 1953.
- [193] Xuereb A., Caelen J., « Un modèle d'interprétation pragmatique en dialogue homme-machine, basé sur la SDRT », *Actes de Traitement Automatique de Langues Naturelles*, Fès, 2004.
- [194] Xuereb A., Caelen J., « Actes de langage et relations rhétoriques en dialogue homme-machine », *Revue de l'université de Moncton, Canada* 36(2) : 5-51, 2005.
- [195] Xuereb A., Caelen J., « Topiques dialogiques », *Actes de Traitement Automatique des Langues Naturelles*, Dourdan, 2005.

Table des matières

Remerciements	i
Résumés	iii
1 Introduction	1
1.1 Introduction	1
1.2 Contexte des recherches	6
1.3 Contrôle pragmatique de la génération	10
1.3.1 Perspective rhétorique sur le dialogue à plusieurs locuteurs	10
1.3.2 Structuration rhétorique en génération des énoncés	11
1.3.3 Contrôle rhétorique des phénomènes linguistiques en génération	15
1.3.4 Efficacité des démarches proposées : illustration	25
1.4 Discussion	27
2 Etat de l'art	29
2.1 Génération pour le dialogue	29
2.1.1 Contexte du problème	29
2.1.2 Génération monologique <i>versus</i> génération dialogique	30
2.2 Approches	32
2.2.1 Position du problème	32
2.2.2 Générateurs à base d'analyse inverse	32
2.2.3 Générateurs textuels à base de grammaires	33
2.2.4 Générateurs textuels à base de structures figées	34
2.2.5 Utilisation des systèmes complets de génération	35
2.3 Génération du langage face au dialogue	36
2.3.1 Position du problème	36
2.3.2 Générateurs à base d'analyse inverse	36
2.3.3 Générateurs textuels à base de grammaires	37
2.3.4 Générateurs textuels à base de structures figées	39
2.3.5 Générateurs « complets » à base de grammaire	40
2.3.6 Générateurs « complets » à base de structures figées	41
2.3.7 Conclusions	42
2.4 Générateurs pour le dialogue	42

2.4.1	Le générateur de Stent	42
2.4.2	Générateur de Jokinen	49
2.5	Conclusions	51
3	Filtrage local des relations rhétoriques	53
3.1	Introduction	53
3.2	Filtrage sémantique	54
3.2.1	Position du problème	54
3.2.2	Emulation de la SDRT dans une logique du premier ordre	55
3.2.3	Mise à jour de la structure de discours dialogique : algorithme	64
3.2.4	Calcul des relations rhétoriques : exemple étendu	65
3.3	Filtrage pragmatique	67
3.3.1	Actes de langage et calcul des relations rhétoriques	67
3.3.2	Calcul des relations rhétoriques : exemple étendu	70
3.4	Conclusions	73
4	Filtrage contextuel des relations rhétoriques	75
4.1	Introduction	75
4.2	Relations rhétoriques et grammaires d'attributs	76
4.2.1	Relations de discours et programmes logiques	76
4.2.2	Aperçu grammatical des relations rhétoriques	77
4.3	Filtrage contextuel	79
4.3.1	Procédure de filtrage « syntaxique »	79
4.3.2	Fondements théoriques	82
4.3.3	Discussion	86
4.3.4	Filtrage numérique des relations rhétoriques	90
4.3.5	Exemple étendu en dialogue	91
4.4	Conclusions	98
5	Structuration rhétorique en dialogue multi-locuteurs	99
5.1	Introduction	99
5.2	Cadre général	100
5.2.1	Conventions	100
5.2.2	Situations de dialogue à plusieurs locuteurs	101
5.3	Dialogue à plusieurs locuteurs	107
5.3.1	Unicité des points de vue	107
5.3.2	Situations en dialogue multi-locuteurs	110
5.3.3	Mise à jour de la structure de discours dialogique	112
5.4	Calcul de la structure rhétorique	117
5.5	Conclusions et perspectives	123

6	Contrôle de l'ellipse sémantique	125
6.1	Introduction	125
6.2	Situations de dialogue à plusieurs	127
6.3	Gestion de l'ellipse sémantique	130
6.3.1	Cas du dialogue traditionnel	130
6.3.2	Extensions au dialogue à plusieurs	134
6.3.3	Discussion	142
6.4	Conclusions et perspectives	144
7	Le degré de force illocutoire	145
7.1	Introduction	145
7.2	Cadre général	147
7.2.1	Structure rhétorique et engagements publics	147
7.2.2	Degré de force illocutoire	149
7.2.3	Situations de dialogue à plusieurs locuteurs	152
7.3	Réglage de la force illocutoire	154
7.3.1	Calcul du degré de force illocutoire	154
7.3.2	Exemples en dialogue homme-machine	162
7.4	Extensions vers le dialogue à plusieurs	165
7.5	Discussion et conclusions	168
8	Génération des connecteurs	173
8.1	Introduction	173
8.2	Génération des connecteurs concessifs en dialogue	175
8.2.1	Relations rhétoriques et connecteurs discursifs	175
8.2.2	Aperçus sémantico-pragmatiques des connecteurs	176
8.2.3	Structure de discours et sémantique des connecteurs	178
8.2.4	Choix des connecteurs : algorithme	180
8.2.5	Traçage sur une situation de dialogue	182
8.3	Ordonnancement argumentatif des énoncés	184
8.3.1	Prémises théoriques	184
8.3.2	Ordonnancement argumentatif des topoï	185
8.3.3	Exemple en dialogue multi-locuteurs	196
8.4	Conclusion et perspectives	197
9	Conclusions	199
9.1	Bilan	199
9.1.1	Conclusions générales	199
9.1.2	Situations délicates en dialogue à plusieurs locuteurs	201
9.2	Développements futurs	203
9.2.1	Contournement des limites et relaxation des contraintes	203
9.2.2	Perspectives à moyen et à long terme	205

A	Degré de force illocutoire et réalisation de surface	211
A.1	Introduction	211
A.2	Les modificateurs déréalisants	212
A.3	Formalisation	213
A.4	Génération superficielle des degrés de force illocutoire	218
B	Glossaire	221
C	La SDRT	229
C.1	Introduction	229
C.2	La SDRT en 1993	229
C.2.1	Position du problème	229
C.2.2	Les éléments définitoires de la SDRT	230
C.2.3	Les relations rhétoriques principales	231
C.3	De nouveaux apports : la SDRT en 2003	233
C.3.1	Extension au dialogue	233
C.3.2	Les remises en cause en dialogue	234
C.3.3	Formalisation de la cohérence discursive	235
C.3.4	La sous-spécification	236
C.3.5	En guise de conclusion : une théorie modulaire	236