



HAL
open science

Syst'eme dialectique multi-agents pour l'aide 'a la concertation

Maxime Morge

► **To cite this version:**

Maxime Morge. Syst'eme dialectique multi-agents pour l'aide 'a la concertation. Système multi-agents [cs.MA]. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2005. Français. NNT : 2005EMSE0014 . tel-00797376

HAL Id: tel-00797376

<https://theses.hal.science/tel-00797376v1>

Submitted on 6 Mar 2013

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.



Système dialectique multi-agents pour l'aide à la concertation

THÈSE.

présentée en vue de l'obtention du titre de

Docteur de l'École Nationale Supérieure des Mines
de Saint Etienne et de l'Université Jean Monnet.

(spécialité informatique)

par

Maxime MORGE

20 juin 2005

Composition du jury :

Rapporteurs :

Philippe MATHIEU Professeur d'Informatique, Université Lille 1

Jean-Paul SANSONNET Directeur de recherche CNRS, LIMSI-CNRS

Examineurs :

Philippe BEAUNE Maître assistant, ENS Mines Saint-Etienne

Olivier BOISSIER Professeur d'Informatique, ENS Mines Saint-Etienne

Jean-Luc KONING Professeur d'Informatique, INPG ESISAR Valence

Nicolas MAUDET Maître de conférences, Université Paris IX

À ma mère.

« La vérité ne se mesure pas à l'applaudimètre. »

Pierre Bourdieu.

Remerciements

À l'issue de mon parcours universitaire, je ressens le besoin de remercier les personnes qui ont joué un rôle pendant ces dix années.

En premier lieu, je remercie les membres du jury. Tout d'abord M. Philippe MATHIEU et M. Jean-Paul SANSONNET de m'avoir fait l'honneur d'accepter d'examiner ce manuscrit. M. Jean-Luc KONING et M. Nicolas MAUDET, d'avoir accepté d'assister à cette soutenance.

Je tiens à remercier tout particulièrement Olivier BOISSIER et Philippe BEAUNE d'avoir accepté d'encadrer ce travail de recherche, pour leur patience et leur ténacité pendant les cinq années au cours desquelles nous avons travaillé ensemble. Chapeau bas chefs !

Un travail de recherche est par essence collectif. Je remercie l'ensemble des membres de l'équipe SMA pour les discussions enrichissantes que nous avons eues : Guillaume Muller, Cosmin Carabelea et Laurent Vercouter. Merci également à Julien, Franck, Rahee, Mahdi, Ben, Oana. . . Je tiens à remercier également Peter McBurney, Frank Dignum, Pavlos Moraitis, Floriana Grasso et Bernard Moulin pour leurs conseils éclairés et leurs encouragements.

Sans financement, il n'y a pas de recherche. Je salue l'Action Concertée Incitative Ville menée par le Ministère de la Recherche au travers de laquelle j'ai obtenu une allocation de recherche. Je remercie en particulier Rémi BAUDOÛI d'avoir coordonné ce projet de recherche pluridisciplinaire. Je salue également le projet « Aide à la Décision et à la Négociation pour la gestion du Territoire » financé en partie par la Région Rhône-Alpes. Je remercie en particulier Didier GRAILLOT d'avoir coordonné ce projet de recherche multi-disciplinaire. Je salue également la Ligue contre le Cancer, l'ENSMSE et le CROUS des bourses qui m'ont été octroyées lorsque j'étais encore étudiant et sans lesquelles j'aurais mis un terme à mes études.

Sans vocation il n'y a pas de recherche. Je salue l'ensemble des membres du département informatique de l'Université Jean Monnet pour la qualité de leurs enseignements : Jean Azema, Pierre Dupont, Philippe Ezequel, Marc Bernard, François Jacquenet, Colin de la Higueira, Christine Largeton et Richard Baron.

Merci à l'ensemble du personnel de l'École des Mines de Saint-Etienne pour leur accueil, notamment au sein du centre G2I/SIMMO aux côtés du centre SITE.

Être exhaustif est impossible, mes remerciements à tous ceux qui m'ont soutenu dans ma quête épistémologique. Amitiés à Claire de Champagne, à Fabienne et Mathias de Valence, à Jérôme et Lolo de l'AGSE, à Olivier de Santiago, à Luc de Radio Dio, à Fabrice et Marie d'Helsinki, à Pom et Mathias de l'Entrepôt, à Momo de la Métare, à Rémi de l'ASEC, à Nono de Roche. Salut à vous jeunesse du monde entier !

Mes derniers mots iront à ma sœur Magali, à ma mère et à ma compagne Karine, meine Liebe.

Résumé

Ce travail de recherche est consacré à la formalisation d'un processus de prise de décision collective et justifiée à l'aide d'un système multi-agents. Ce système est constitué d'agents cognitifs, autonomes et sociaux en interaction. S'appuyant sur la communication par envoi de messages, on parle de modèle d'interaction directe entre agents. Dans ce manuscrit, nous défendons la thèse selon laquelle le dialogue peut être considéré comme un processus argumentatif au travers duquel les participants jouent et arbitrent pour aboutir à une décision collective. Nous proposons ici un modèle de dialogue entre agents. Ce modèle, intitulé DIAL (*DIALogue Is an Argumentative Labour*), sert de base à un système d'aide à la concertation.

The goal of this research work is the formalization of a justified and collective decision by a Multi-Agent System (MAS). Such a system consists of autonomous, cognitive and social agents communicating by messages. In this thesis, we assert that the dialogue could be considered as an argumentative process in which parties play and arbitrate to reach an agreement. We propose here a dialectics multiagent system to formalize such a process. This system, entitled DIAL (DIALogue Is an Argumentative Labour), is used to provide a group decision support system.

Table des matières

I	Etat de l’art	7
1	Modèle d’interaction directe entre agents	9
1.1	Introduction	9
1.2	Théorie des actes de langage	10
1.2.1	Langage : de la représentation à l’action	11
1.2.2	Typologie des actes	12
1.3	Langage de communication d’agents	14
1.3.1	Syntaxe	15
1.3.2	Sémantique	16
1.3.3	Pragmatique	16
1.4	Protocole d’interaction	18
1.4.1	Langage semi-formel	19
1.4.1.1	Formalisme	19
1.4.1.2	Exemple	20
1.4.1.3	Limites	21
1.4.2	Langage formel	21
1.4.2.1	Formalisme	21
1.4.2.2	Exemple	24
1.4.2.3	Limites	25
1.5	Synthèse	25
2	Argumentation	29
2.1	Introduction	29

2.2	Argument	30
2.2.1	Motivation	30
2.2.2	Structure des arguments	31
2.2.3	Force des arguments	32
2.3	Systèmes d'argumentation abstraits	33
2.3.1	Système d'argumentation	34
2.3.2	Système d'argumentation à base de préférences	39
2.3.3	Système d'argumentation à base de valeurs	41
2.4	Logiques argumentatives	44
2.4.1	Relation de contradiction	45
2.4.2	Relation de priorité statique	48
2.4.3	Relation de priorité dynamique	52
2.5	Synthèse	57
3	Dialectique	59
3.1	Introduction	59
3.2	Système dialectique	60
3.3	Catégorisation des dialogues	64
3.4	Jeu logique d'évaluation sémantique	65
3.5	Jeu argumentatif d'évaluation sémantique	72
3.6	Synthèse	79
4	Modèle de dialogue entre agents	81
4.1	Introduction	81
4.2	Dialogue et engagements	82
4.2.1	Dialogue pour la formation d'équipe	82
4.2.1.1	Cadre flexible de résolution collaborative de problème	82
4.2.1.2	Formalisation des stratégies d'engagements	83
4.2.2	DIAGAL	85
4.2.2.1	Engagement social	85
4.2.2.2	Jeu de dialogue	86

4.3	Dialogue et arguments	88
4.3.1	Système multi-agents argumentatifs	88
4.3.2	Système dialectique multi-agents	91
4.3.2.1	Cadre	91
4.3.2.2	Protocole de discussion	95
4.3.2.3	Protocole naïf	101
4.3.2.4	Protocole bavard	103
4.4	Modèle de dialogue	108
4.4.1	Le modèle de dialogue	108
4.4.2	Les relations entre engagements et arguments	109
4.5	Synthèse	111
5	Conclusion de l'état de l'art	115
5.1	Modèles de raisonnement argumentatif	115
5.2	Modèles de dialogue entre agents	117
II	Modèle DIAL	119
6	Introduction du modèle	121
7	Système d'argumentation multi-agents	123
7.1	Introduction	123
7.2	Langage sous-jacent	124
7.3	Arguments	127
7.4	Relations entre arguments	129
7.4.1	Relation d'attaque	129
7.4.2	Relations de défaite	130
7.5	Acceptabilité des arguments	132
7.6	Synthèse	134

8	Système multi-agents argumentatifs	137
8.1	Introduction	137
8.2	Langages communs	138
8.2.1	Langage de représentation des connaissances	138
8.2.2	Langage de communication d'agents	138
8.3	Raisonnement conjoint	139
8.3.1	Agents argumentatifs	139
8.3.2	Théorie argumentative étendue	143
8.3.3	Sémantique sociale	145
8.3.3.1	Engagements sociaux	145
8.3.3.2	Principe coopératif d'adoption des hypothèses	146
8.3.4	Sémantique argumentative	148
8.3.4.1	Conditions rationnelles d'énonciation	148
8.3.4.2	Tactiques d'énonciation	150
8.4	Synthèse	152
9	Système dialectique multi-agents	155
9.1	Introduction	155
9.2	Coup dialogique	156
9.3	Système dialectique	157
9.4	Protocole de dialogue	162
9.5	Règles de séquence	164
9.5.1	Réponse à une question	165
9.5.2	Réponse à une affirmation	166
9.5.3	Réponse à une mise en doute	170
9.5.4	Terminaison	170
9.6	Jeu de dialogue	171
9.6.1	Demande d'information	171
9.6.2	Persuasion	175
9.7	Synthèse	177

10 Conclusion du modèle	181
III Application	183
11 Système d'aide à la concertation	185
11.1 Introduction	186
11.2 Motivation	186
11.3 Principe	188
11.4 Logique argumentative	189
11.4.1 Représentation du problème	189
11.4.2 Expressions des préférences	190
11.4.3 Synthèse des jugements	192
11.4.4 Théorie argumentative décisionnelle	194
11.5 Système multi-assistants	197
11.6 Système dialectique multi-assistants	198
11.6.1 Jeu de soumission d'activité	198
11.6.2 Jeu de persuasion	198
11.7 Aide à la concertation	199
11.7.1 Élaboration conjointe d'un schéma argumentatif	199
11.7.2 Détection de conflit et de consensus	200
11.8 Implémentation	201
11.8.1 Architecture	201
11.8.2 Interface homme-machine	202
11.8.3 Modèle de raisonnement	203
11.8.4 Modèle d'agent	203
11.9 Synthèse	204
IV Conclusions & Perspectives	211
12 Conclusion	213
12.1 Problématique	213

12.2 Contributions	214
12.3 Limites	215
12.4 Perspectives	215
V Références et index	217
VI Annexes	227
A Logique classique	229
A.1 Introduction	229
A.2 Syntaxe	230
A.2.1 Langage propositionnel	230
A.2.2 Langage du premier ordre	230
A.3 Sémantique	232
A.3.1 Logique propositionnelle	233
A.3.2 Logique du premier ordre	234
A.4 Système de preuve	236
A.4.1 Logique propositionnelle	237
A.4.2 Logique du premier ordre	237
A.5 Synthèse	237
B Complexité computationnelle	239
B.1 Introduction	239
B.2 Algorithme	240
B.3 Analyse algorithmique	240
B.4 Problème de décision	242
B.4.1 Réduction algorithmique	243
B.4.2 Complexité en espace	244
B.5 Problème de recherche	245
B.6 Synthèse	247

C	Relation d'ordre	249
C.1	Introduction	249
C.2	Relations binaires	250
C.3	Relations d'ordre	250
C.4	Conclusion	252
D	Preuves	255
D.1	Introduction	255
D.2	Terminaison d'une demande d'information	255
D.3	Correction d'une demande d'information	256
D.4	Complétude d'une demande d'information	258
D.5	Terminaison des persuasions	259
D.6	Correction d'une persuasion	260
E	Exemples	263
E.1	Introduction	263
E.2	Demande d'information correcte et complète	263
E.3	Demande d'information correcte et incomplète	269
E.4	Persuasion	271

Table des figures

0.0.0.I	Organisation du manuscrit	6
1.1.0.I	Schéma d'un système informatique communiquant selon Claude Shannon	10
1.2.1.I	Syllogisme écrit en langage naturel et traduit en logique des prédicats	11
1.3.1.I	Grammaire DTD d'un message FIPA-ACL	15
1.4.1.I	Protocole FIPA « <i>Contract-Net</i> »	27
1.4.2.I	Représentation d'un automate	28
1.4.2.II	Protocole de marchandage	28
2.2.2.I	Schéma de l'argumentation selon Stephen Toulmin	32
2.3.1.I	Représentation sous forme de graphe orienté du système d'argumentation AS	34
2.3.1.II	Algorithme pour la construction d'une extension raisonnable d'un système d'argumentation	36
2.3.1.III	Représentation sous forme de graphe orienté des systèmes d'argumentation AS ₁ (à gauche) et AS ₂ (à droite)	36
2.3.1.IV	Représentation sous forme de graphe orienté des systèmes d'argumentation AS ₂ (à gauche) et AS ₃ (à droite)	37
2.3.2.I	Représentation du système d'argumentation à base de préférences AS ₄	41
2.3.3.I	Représentation du système d'argumentation AAS ₅ spécifique à une audience α	44
2.4.1.I	Diagramme de Hasse des relations d'attaque	47
2.4.2.I	Représentation du système d'argumentation à base de préférences	51

2.4.3.I	Représentation d'une théorie argumentative	56
3.4.0.I	Représentation sous forme normale extensive du jeu logique d'évaluation sémantique de la formule Φ dans l'interprétation M . . .	70
3.5.0.II	Représentation sous forme de graphe orienté du système d'argumentation AS_2	78
3.5.0.III	Représentation sous forme normale extensive des confrontations sur P (à gauche) et Q (à droite) dans le système d'argumentation AS_2	78
4.2.2.I	Cycle de vie d'un engagement	86
4.3.2.I	Représentation sous forme de graphe orienté du système d'argumentation AS_2	101
4.3.2.II	Représentation sous forme normale extensive du système dialectique $DS_{\text{Moves}}(\psi_0, \text{Argbase}(h))$	102
4.3.2.III	Représentation sous forme de graphe orienté du système d'argumentation AS_5	104
4.3.2.IV	Représentation du dialogue orienté par un protocole bavard à réponses uniques.	104
4.3.2.V	Représentation sous forme de graphe orienté du système d'argumentation AS_6	105
4.3.2.VI	Représentation d'un dialogue orienté par un protocole bavard à réponses multiples.	105
4.3.2.VII	Représentation du dialogue orienté par un protocole libre.	108
4.4.1.I	Liens entre architecture d'un agent dialogique et modèle de dialogue	110
4.4.2.I	Cycle de vie du contenu d'un engagement	113
5.1.0.I	Arbre de décision pour classer les modèles de raisonnement argumentatif	116
5.2.0.II	Arbre de décision pour classer les modèles de dialogue entre agents.	117
7.2.0.I	Théories argumentatives mono-agent	127
7.4.2.I	Relations entre arguments selon l'agent ag_1	131
7.4.2.II	Relation entre arguments selon l'agent ag_2	132

7.5.0.III	Représentation des théories argumentatives \mathcal{AT}_{\cup_A} (au centre), \mathcal{AT}_1 (à gauche) et \mathcal{AT}_2 (à droite)	133
8.3.1.I	Théorie argumentative associée à l'agent ag_1	142
8.3.1.II	Théorie argumentative associée à l'agent ag_2	142
8.3.2.I	Théorie argumentative étendue associée à l'agent ag_2	145
9.5.1.I	Réponse à une question	166
9.5.2.I	Réponse à une affirmation selon la règle d' « Affirmation/Accueil »	168
9.5.2.II	Réponse à une affirmation selon la règle d' « Affirmation/Réfutation »	169
9.6.1.I	Jeu de demande d'information	179
9.6.2.I	Jeu de persuasion	180
11.3.0.I	Principe du système d'aide à la concertation	188
11.4.1.I	Hierarchie décisionnelle pour la localisation d'ITER	190
11.4.3.I	Hierarchie décisionnelle évaluée pour la localisation d'ITER . . .	195
11.6.1.I	Jeux de soumission d'activité sous forme normale extensive . . .	198
11.7.1.I	Diffusion d'un nouveau critère c_l	199
11.6.2.I	Jeu de persuasion dans une hiérarchie décisionnelle	205
11.7.1.II	Hierarchie décisionnelle conjointe pour la localisation d'ITER . .	206
11.7.2.I	Hierarchie décisionnelle évaluée de l'initiateur et du partenaire . .	206
11.7.2.II	Persuasion entre deux agents argumentatifs décisionnels	207
11.8.1.I	Architecture du système d'aide à la concertation	207
11.8.2.I	Capture d'écran de l'interface graphique proposée par JAHP . .	208
11.8.3.I	Théorie argumentative mono-agent	208
11.8.4.I	Règles de réponses à une question	209
A.3.1.I	Tables de vérité des connecteurs logiques	233
B.4.2.I	Diagramme de Venn des classes des problèmes de décision	245
B.5.0.II	Diagramme de Venn de la hiérarchie polynomiale	247
E.2.0.I	Théorie argumentative étendue de l'initiateur dans la situation initiale pour la première demande d'information	264

E.2.0.II	Théorie argumentative étendue du partenaire dans la situation initiale pour la première demande d'information	264
E.2.0.III	Théorie argumentative étendue du témoin dans la situation initiale pour la première demande d'information	265
E.2.0.IV	Théorie argumentative de l'agent omniscient	265
E.2.0.V	Théorie argumentative étendue du témoin au terme de la première demande d'information	268
E.3.0.VI	Théorie argumentative de l'initiateur dans la situation initiale pour la seconde demande d'information	269
E.3.0.VII	Théorie argumentative de l'agent omniscient de la seconde demande d'information	270
E.4.0.VIII	Théorie argumentative étendue de l'initiateur dans la situation initiale pour la persuasion	271
E.4.0.IX	Théorie argumentative étendue du partenaire dans la situation initiale pour la persuasion	272
E.4.0.X	Théorie argumentative étendue du témoin au terme de la persuasion qui considère l'initiateur plus compétent que le partenaire	276
E.4.0.XI	Théorie argumentative étendue du témoin au terme de la persuasion qui considère l'initiateur moins compétent que le partenaire	277

Liste des tableaux

2.3.1.I	Complexité algorithmique des problèmes de décision liés au système d'argumentation	38
2.4.2.I	Complexité algorithmique des problèmes de décision liés à la relation de priorité statique	53
3.2.0.I	Mise à jour des tableaux d'engagements au cours du dialogue . .	63
3.3.0.II	Aperçu systémique des types de dialogues	64
3.3.0.III	Synthèse de la catégorisation des dialogues	66
4.2.2.I	Conditions et règles du jeu de requête	87
4.2.2.II	Classification des jeux de dialogue utilisés par DIAGAL	88
4.3.2.I	Ensemble des actes de langage et leurs réponses potentielles . . .	92
9.5.1.I	Règle de « Question/Réponse » : question et réponses potentielles.	165
9.5.2.I	Règle d' « Affirmation/Accueil » : affirmation et réponses potentielles.	166
9.5.2.II	Règle d' « Affirmation/Réfutation » : affirmation et réponses potentielles.	167
9.5.3.I	Règle de « Mise en doute/Réponse » : mise en doute et réponses potentielles.	170
9.5.4.I	Règle de terminaison.	170
9.6.1.I	Règles de séquence du protocole de demande d'information à réponses uniques.	172
9.6.2.I	Règles de séquence du protocole de persuasion à réponses uniques.	176
11.4.2.I	Échelle de comparaisons binaires	191

11.4.2.II	Matrice de comparaison binaire A_{c_1} et vecteur de priorités associé W_{c_1}	192
11.4.3.I	Indices de cohérence pour des matrices aléatoires	192
12.3.0.I	Aperçu systémique des types de dialogues qui ont (ou non) été abordés	215
C.3.0.I	Relations d'ordre	253

Introduction

Dans ce chapitre nous présenterons la problématique générale de la thèse. À la suite de l'exposé des motivations de ce travail de recherche, nous le situerons dans un contexte scientifique, en l'occurrence les Systèmes Multi-Agents (SMA), et en précisons les objectifs. Enfin nous détaillerons la structure de ce manuscrit.

Motivation

Dans le monde réel, la plupart des décisions implique un large panel d'acteurs dont l'adhésion influe fortement sur leur succès. Le processus de décision doit ainsi donner aux acteurs la conviction que la décision est équitable (*fair effect process*). Ce constat est à l'origine d'un changement de perspective dans l'élaboration des procédures démocratiques en insistant sur les dimensions collectives et discutées des décisions.

Dans le domaine de la décision publique, on peut distinguer deux modalités de formation d'une volonté générale : la démocratie représentative et la démocratie dialogique. La démocratie représentative consiste en un processus d'agrégation des préférences individuelles à travers lequel les profanes et les citoyens délèguent leur pouvoir aux élus et aux experts. La démocratie dialogique est un processus participatif de composition des perspectives et des intérêts au travers duquel la société civile débat et délibère. Les forums hybrides, les conférences de consensus et les groupes de discussion, qu'ils portent sur des controverses socio-techniques (OGM, ESB, thérapie génique, téléphonie mobile, ...) ou sur des projets territoriaux (autoroute, centrale nucléaire, gestion de l'eau, ...), constituent des expérimentations de démocratie dialogique.

Dans le cadre de notre travail, nous nous intéressons à cette dernière modalité. Nous souhaitons fournir un outil informatique d'aide à la concertation pour que les acteurs puissent collaborer à distance et de manière asynchrone. Les décisions prises jusqu'alors à huis clos et dont la justification restait obscure deviennent, *via* l'outil informatique, ouvertes et transparentes. Ce changement de perspective permet, d'un point de vue social et politique, de tirer parti de l'élargissement du cercle de

concertation afin d'en enrichir l'expertise et de prévenir d'éventuels blocages, et du point de vue de la prise de décision rationnelle, d'exploiter la créativité des différents acteurs au cours de la phase de formulation du problème.

Ce travail de recherche est consacré à la formalisation d'un processus de prise de décision concertée afin d'élaborer un outil informatique d'aide à la concertation.

Contexte scientifique

Considérant que la prise de décision est un processus multi-acteurs, notre étude se situe dans le cadre des systèmes multi-agents. Nous nous intéressons plus particulièrement à des agents cognitifs, autonomes et sociaux en interaction.

On désigne ici par **agent autonome**, un programme informatique qui représente un acteur, qui le modélise ou qui l'assiste. Souhaitant modéliser les mécanismes de raisonnement évolués des acteurs, nous considérons des **agents cognitifs**, *i.e.* des agents capables de manipuler des représentations symboliques et explicites. Le processus de décision étant collaboratif, nous considérons des **agents sociaux**, *i.e.* des agents qui coopèrent avec d'autres agents dans un SMA.

Le processus de délibération est le résultat de l'activité psychologique individuelle réalisée par les acteurs pour atteindre leurs buts. En conséquence, nous considérons ici les modèles de **coordination subjective**. En d'autres termes, les agents sont considérés comme des entités coordinantes, les sujets de la coordination et non pas comme des entités coordonnées c'est-à-dire les objets de la coordination.

En résumé, notre travail se focalise sur les modèles de coordination subjective entre agents cognitifs, autonomes et sociaux. S'appuyant sur la communication par envoi de messages, ils sont qualifiés de modèles d'interaction directe entre agents.

Objectifs et contributions

Cette thèse contribue à l'étude des modèles de dialogue entre agents. Ces modèles d'interaction directe entre agents logiciels s'inspirent de la dialectique. Par dialectique, on désigne l'étude ou la conception de dispositifs pour la mise en situation de l'argumentation.

Nous défendons la thèse selon laquelle le dialogue peut être modélisé comme un processus argumentatif au travers duquel les participants jouent et arbitrent pour aboutir à une décision collective. Nous proposons un modèle de dialogue entre agents

qui sert de base à l'élaboration d'un outil informatique d'aide à la concertation. Cet objectif se décompose en deux sous-objectifs :

- *Modèle de dialogue entre agents.* La définition d'un cadre formel dans lequel deux agents dialoguent pour atteindre un accord. Le raisonnement argumentatif des agents leur permet de gérer les interactions entre des arguments contradictoires que chacun considère comme plus ou moins forts. Les agents justifient les hypothèses sur lesquelles ils s'engagent et prennent en compte les engagements de leur interlocuteurs en fonction de la réputation de l'agent qui a transmis cette information. Un agent tiers, responsable de la décision finale, rend son arbitrage non seulement en fonction des arguments avancés mais également en fonction de son estimation de la compétence relative de chacun des deux joueurs.
- *Système d'aide à la concertation.* La définition d'un outil informatique qui fournit d'une part des fonctionnalités pour l'élaboration collaborative d'une représentation du problème et qui fournit d'autre part des fonctionnalités pour l'élucidation des consensus et des conflits.

Organisation du manuscrit

Ce manuscrit est organisé en trois parties. La première partie dresse un état de l'art des modèles de dialogue entre agents. La seconde partie décrit notre modèle. La troisième partie aborde sa mise en œuvre. L'agencement des chapitres centraux est représenté dans la figure 0.0.0.I.

Partie I : État de l'art. Cette partie dégage les caractéristiques essentielles que doit posséder un modèle de dialogue entre agents pour formaliser une décision collective et débattue. Dans cette perspective, cette partie met en exergue les limites de l'approche mentalistique et précise comment l'approche dialectique y répond.

Modèle d'interaction directe entre agents. Dans ce chapitre, nous montrons que pour interagir, des agents logiciels doivent partager un langage commun, appelé langage de communication d'agents (ACL). En s'inspirant de la théorie des actes de langages, l'approche mentalistique présentée dans ce chapitre spécifie la pragmatique des messages en terme d'état mentaux. Pour les trois raisons suivantes, cette approche ne permet pas de modéliser une prise de décision concertée :

1. Les messages constituent des briques élémentaires qui doivent être ordonnées dans une conversation, conformément à un protocole d'interaction. D'une part, ces protocoles sont trop rigides. D'autre part, les propriétés d'un protocole ne permettent ni de garantir l'obtention d'un résultat ni de préjuger de sa qualité.

2. La sémantique des actes de langages est privée et subjective. Un agent qui assiste à une conversation ne confère pas le même sens à un acte de langage que son destinataire. *A fortiori*, il ne réalise pas les mêmes inférences.
3. Le modèle de raisonnement sous-jacent ne permet pas à un agent de gérer les incohérences entre ses propres croyances et les informations reçues.

Pour pallier cette dernière lacune, on envisage un modèle de raisonnement argumentatif.

Argumentation. Dans ce chapitre, nous montrons en quoi l'argumentation constitue un modèle de raisonnement adapté au processus cognitif d'un agent autonome et social pour gérer les interactions entre les arguments internes qui explicitent ses croyances et les arguments externes qui créditent des croyances contradictoires provenant d'autres agents.

Néanmoins, l'argumentation ne constitue pas un modèle satisfaisant pour étudier la dynamique des flux informationnels dans un processus d'argumentation. C'est pourquoi nous envisageons les travaux relatifs à la dialectique.

Dialectique. Dans ce chapitre, nous définissons la dialectique, qu'elle soit descriptive ou formelle, comme l'étude ou la conception de dispositifs au travers desquels l'argumentation est mise en situation. Un système dialectique est un dispositif au travers duquel un ensemble de participants jouent des coups dialogiques. D'une part, un dialogue est régulé par une convention, *i.e.* un ensemble de règles dialectiques. Ces règles spécifient, étant donné un contexte dialogique, les locutions qui sont autorisées (ou non). D'autre part, les déclarations sont stockées au fur et à mesure dans des structures de données intitulées tableaux d'engagements.

Ayant dressé le cadre général dans lequel peut se situer notre travail, nous examinons les différents travaux abordant une problématique semblable à la notre.

Modèle de dialogue entre agents. Dans ce chapitre, nous présentons les modèles d'interaction directe entre agents qui s'inspirent de la dialectique. Contrairement à l'approche mentalistique, cette approche permet de modéliser une prise de décision concertée :

1. *L'approche conventionnelle.* Les règles dialectiques constituent une alternative au protocole d'interaction. D'une part, ce sont des structures collaboratives de communication flexible qui permettent de coordonner l'activité dialogique des agents. D'autre part, le système dialectique multi-agents constitue un cadre formel qui permet de garantir l'obtention d'un résultat et d'évaluer sa qualité.
2. *La sémantique sociale des actes de langage.* Les engagements pris sont publics et objectifs. Un agent qui assiste au dialogue confère le même sens aux

actes de langages et réalise les mêmes inférences qu'il en soit le destinataire ou non.

3. *La sémantique argumentative des actes de langage.* Le raisonnement des agents s'appuie sur une logique argumentative. Ce modèle de raisonnement permet aux agents de gérer les incohérences entre ses propres croyances et les informations transmises par d'autres agents.

Les caractéristiques essentielles que doit posséder un modèle de dialogue entre agents ayant été dégagées, nous sommes en mesure de réaliser une synthèse des travaux existants et de proposer notre propre modèle.

Partie II : DIAL. Cette partie décrit le modèle DIAL, un modèle de dialogue entre agents qui permet de formaliser une prise de décision concertée. Suite aux exigences identifiées dans la partie précédente, nous proposons ici un modèle de dialogue entre agents qui y répond.

Système d'argumentation multi-agents. Nous définissons dans ce chapitre la logique argumentative qui constitue le modèle de raisonnement des agents. Cette logique argumentative permet de gérer les interactions entre des arguments contradictoires qui sont plus ou moins forts selon l'agent envisagé.

Dans un tel système d'argumentation, les connaissances sont communes. À l'inverse, les connaissances sont distribuées dans un système multi-agents argumentatifs.

Système multi-agents argumentatifs. Nous présentons dans ce chapitre les éléments du modèle DIAL qui permettent aux agents d'échanger leurs connaissances et de raisonner conjointement. Les agents argumentatifs font non seulement référence à des échelles de valeurs différentes, mais ils ont aussi des connaissances différentes.

Dans un tel système multi-agents argumentatifs, les connaissances sont échangées mais les convictions des agents diffèrent. À l'inverse, un système dialectique multi-agents permet d'aboutir à un accord.

Système dialectique multi-agents. Nous présentons dans ce chapitre les éléments du modèle DIAL qui permettent de gérer le flux d'information en régulant l'enchaînement des messages afin de mener à bien l'interaction.

Le modèle DIAL ayant été décrit, nous sommes en mesure de synthétiser cette contribution et d'en proposer une application.

Partie III : Application. Cette partie décrit un outil informatique d'aide à la concertation qui s'appuie sur le modèle DIAL.

Système d'aide à la concertation. Le système proposé dans ce chapitre fournit des outils pour l'élaboration collaborative des schémas argumentatifs et pour élucider les consistances et les inconsistances entre les préférences des acteurs et donc détecter les consensus et les conflits.

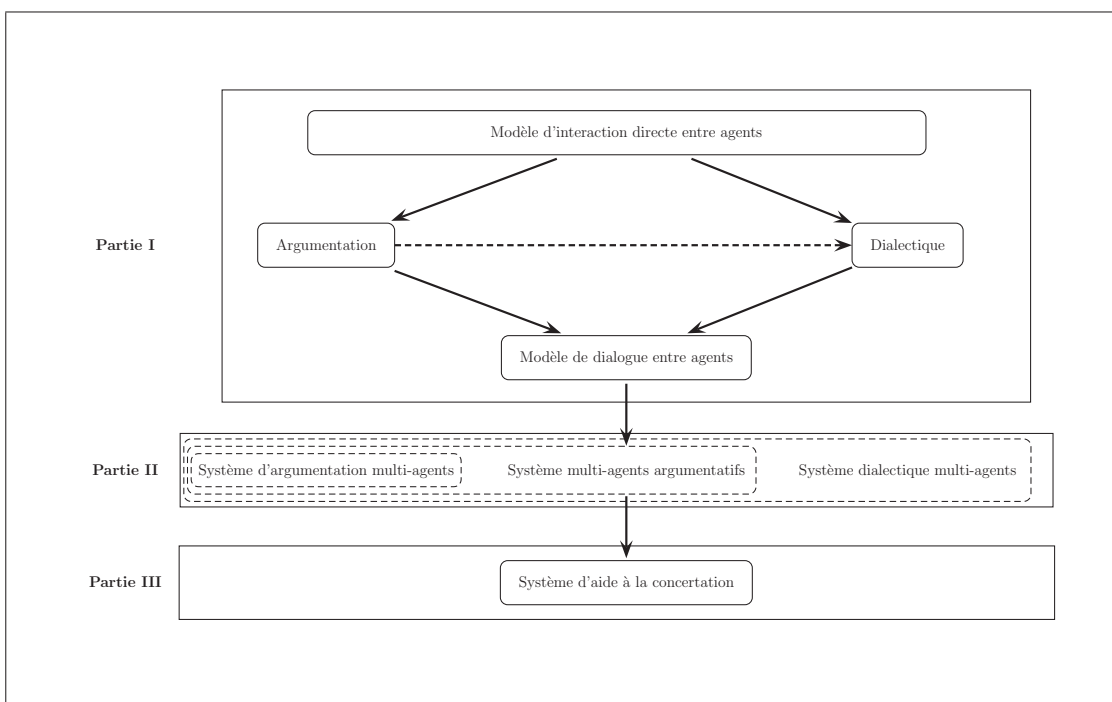


Fig. 0.0.0.I: Organisation du manuscrit

Première partie

Etat de l'art

“When I use a word,” Humpty Dumpty said, in rather a scornful tone, “it means just what I choose it to mean – neither more nor less.” “The question is,” said Alice, “whether you can make words mean so many different things.”

Alice Through the Looking Glass
by Lewis Carroll

Chapitre 1

Modèle d’interaction directe entre agents

Sommaire

1.1	Introduction	9
1.2	Théorie des actes de langage	10
1.2.1	Langage : de la représentation à l’action	11
1.2.2	Typologie des actes	12
1.3	Langage de communication d’agents	14
1.3.1	Syntaxe	15
1.3.2	Sémantique	16
1.3.3	Pragmatique	16
1.4	Protocole d’interaction	18
1.4.1	Langage semi-formel	19
1.4.2	Langage formel	21
1.5	Synthèse	25

1.1 Introduction

La plupart des systèmes de communication entre entités informatiques s’appuie sur le modèle de Claude Shannon [60]. Dans ce modèle, la communication est considérée comme la reproduction exacte ou approximative d’une information entre un émetteur et un récepteur.

Selon Claude Shannon, un système informatique communiquant (cf figure 1.1) est constitué de cinq éléments :

1. une source d’information qui produit le message ;

2. un émetteur qui encode voire compresse le message dans le but de générer un signal ;
3. un canal de communication, éventuellement bruité, qui transmet le signal ;
4. un récepteur qui décode voire décompresse le signal dans le but de reconstruire le message transmis ;
5. un destinataire de l'information qui reçoit le message.

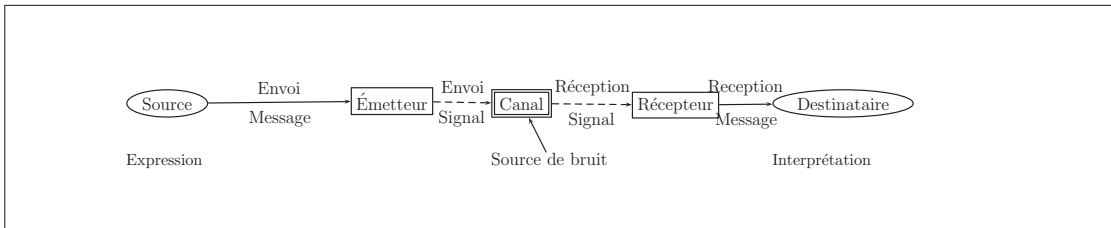


Fig. 1.1.0.I: Schéma d'un système informatique communiquant selon Claude Shannon

Cette approche, bien qu'ayant prouvé son efficacité, est insuffisante pour la définition d'un modèle d'interaction directe entre agents. En effet, la sémantique des messages n'est pas considérée comme pertinente par Claude Shannon. Or cette dimension est essentielle pour la génération et l'interprétation automatique de messages par des agents logiciels. Pour communiquer, les agents ne doivent pas seulement lire les messages, ils doivent comprendre ces messages et savoir les manipuler.

Dans cette optique, des langages de communication d'agents (*ACL*, *i.e.* *Agent Communication Language*) ont été définis pour automatiser la génération, l'interprétation voire le séquençement des messages. Ce chapitre est dédié à leur présentation.

Dans un premier temps, nous allons présenter dans la section 1.2 la théorie des actes de langage qui est le fondement de la plupart des langages de communication d'agents. Nous présenterons ensuite dans la section 1.3 le langage le plus répandu : FIPA-ACL. Nous terminerons ce chapitre par la présentation des protocoles d'interactions qui permettent d'enchaîner les échanges de messages (cf section 1.4).

1.2 Théorie des actes de langage

La théorie des actes de langage est une théorie de la communication fondée dans les années 60 par Austin¹ [6] et développée par Searle² [58] ainsi que Vanderveken³ [64, 63]. Cette théorie affirme que la production d'un énoncé s'assimile à l'ac-

¹John Langshaw Austin (1911-1960), philosophe anglais.

²John R Searle, professeur de philosophie à Berkeley, Californie.

³Daniel Vanderveken, professeur de philosophie à Trois-Rivières, Québec.

complissement d'une action. En d'autres termes, parler c'est agir! Cette théorie permet d'identifier les constituants élémentaires de la communication. C'est la raison pour laquelle, elle a été d'une grande influence dans le domaine des systèmes multi-agents.

Cette théorie met en exergue les énonciations qui réalisent une action (cf section 1.2.1) et en propose une typologie (cf section 1.2.2).

1.2.1 Langage : de la représentation à l'action

La philosophie du langage [68] est une partie de la philosophie qui porte sur la signification ou le sens en général. Elle associe le sens d'une phrase à la représentation d'un état des choses qui peut être vraie ou fausse. Cette philosophie du langage « idéal » considère le langage ordinaire comme confus, simpliste, et rempli d'erreurs. Il doit être corrigé dans une version formelle plus rigoureuse et sans ambiguïté, conformément à la logique.

À l'inverse, la philosophie du langage ordinaire [6] introduite par John Austin prétend éviter les excès de formalisme pour donner plus d'attention aux usages et aux pratiques. Elle attaque le représentationalisme sur lequel se fonde toute la philosophie du langage. Elle prend le contre-pied de la théorie précédente en mettant en cause la fonction descriptive du langage. Cette nouvelle théorie isole un phénomène évident mais auquel la philosophie du langage n'a pas accordé suffisamment d'attention : certains énoncés ne reflètent pas la réalité. L'exemple suivant illustre ce phénomène.

Exemple n° 1.2.1.I [Énoncé et réalité]

Dans la figure 1.2.1.I, nous avons reproduit un syllogisme bien connu exprimé en langage naturel. D'après la philosophie du langage, la signification de ce syllogisme est assimilable à un ensemble de formules de la logique des prédicats. Cette idéalisation nous permet de confirmer que Socrate est mortel.

Socrate est un homme	$\text{homme}(\text{Socrate})$
Tous les hommes sont mortels	$\forall x \text{ homme}(x) \rightarrow \text{mortel}(x)$
<hr/> Socrate est mortel	<hr/> mortel(Socrate)

Fig. 1.2.1.I: Syllogisme écrit en langage naturel et traduit en logique des prédicats

Une telle analyse est réductrice. Que peut-on dire du sens associé à chacune des phrases suivantes : "Passe-moi une clef de 12!", "Je vous déclare pacsés.", "Je m'engage à me présenter à un concours de recrutement de l'enseignement supérieur, conformément au

dispositif de l'article 2 du décret n° 88-654 du 7 mai modifié."? Sont-elles vraies ou fausses?

Pour pallier cette lacune, Austin [6] considère qu'une énonciation vise à accomplir une action pour modifier l'état du monde. Cet acte intentionnel est appelé **acte de langage**. Austin s'est efforcé de mettre en évidence les éléments spécifiques à de tels actes en distinguant :

- la valeur de **locution** : la signification de la production d'une suite de signes. Cette composante se rapporte à la formulation d'un énoncé, *i.e.* ce qui est dit ;
- la valeur d'**illocution** : l'intention exprimée par l'action réalisée. Cette composante désigne l'action effectuée lors de la formulation de l'énoncé, *i.e.* l'acte réalisé par le fait de le dire ;
- la valeur de **perlocution** : l'effet obtenu par la suite de signes sur l'interlocuteur. Cette composante se rapporte aux conséquences sur les états mentaux de l'interlocuteur, *i.e.* le but dans lequel cet acte a été réalisé ;

La valeur illocutoire, c'est-à-dire l'intention ainsi exprimée, est présentée par Austin comme la caractéristique essentielle de l'acte. C'est pour cette raison qu'il est d'usage de parler d'**acte illocutoire** en lieu et place d'acte de langage.

Exemple n° 1.2.1.II [Acte de langage] *Reprenons l'une des phrases qui n'a pu être analysée dans l'exemple précédent.*

La locution "Passe moi une clef de 12!" a une valeur illocutoire : l'émetteur donne un ordre. Ce qui est dit a un effet perlocutoire : l'interlocuteur transmet ou du moins a l'intention de transmettre la clef de 12.

On distingue ainsi deux catégories d'actes illocutoires. D'une part, les **constatifs** sont des énoncés dont la vérité dépend de l'état du monde. D'autre part, les **performatifs**, ceux qui accomplissent une action, sont des énoncés dont la vérité dépend du fait qu'ils ont (ou non) été prononcés. Les performatifs ne sont pas uniquement des ordres. Une typologie permet de classer ces performatifs.

1.2.2 Typologie des actes

La théorie des actes de langage tente de classer les verbes selon l'acte illocutoire qu'ils accomplissent. Searle [58] propose les cinq catégories suivantes de performatifs :

1. les actes **assertifs** expriment une affirmation : observer, informer, témoigner, démentir ;
2. les actes **commissifs** engagent le locuteur dans une action : promettre, souhaiter, menacer ;
3. les actes **directifs** engagent l'interlocuteur dans l'exécution d'une action : demander, supplier, questionner, ordonner, conseiller ;

4. les actes **déclaratifs** réalisent une action par leur seule énonciation : définir, condamner, ratifier ;
5. les actes **expressifs** expriment l'état mental du locuteur : s'excuser, remercier, féliciter, récriminer.

La théorie des actes de langage est une théorie de la signification et de la communication qui analyse la forme logique des actes illocutoires. Les actes sont de la forme : $F(P)$, où P est le **contenu propositionnel** c'est-à-dire la composante représentationnelle de l'acte et où F représente la **force illocutoire** c'est-à-dire la composante intentionnelle de l'acte.

Les actes illocutoires sont des actions intentionnelles. Toute tentative de les accomplir peut réussir ou échouer. Les actes illocutoires ont donc des conditions de succès et de satisfaction.

Les **conditions de succès** d'un acte illocutoire sont les conditions qui doivent être remplies dans un contexte d'énonciation pour que le locuteur réussisse à y accomplir cet acte. Il faut d'abord réussir à dire ce que l'on veut affirmer, promettre, déclarer, demander ou exprimer. Ensuite, il faut que le contexte de l'énonciation soit approprié pour l'accomplissement de l'acte illocutoire visé. Par exemple, la promesse qui consiste à se présenter à un concours de recrutement de l'enseignement supérieur n'est manifestement pas tenue si l'auteur ne soutient pas sa thèse.

Afin de couvrir toutes les forces illocutoires, la notion théorique de satisfaction vient généraliser la notion classique de vérité. Les **conditions de satisfaction** d'un acte illocutoire sont les conditions qui doivent être remplies pour qu'il y ait correspondance entre les mots et l'état du monde.

En accomplissant un acte illocutoire, le locuteur relie son contenu propositionnel au monde avec l'intention d'établir une certaine correspondance entre les mots et les choses selon une **direction d'ajustement**. La typologie searlienne met les cinq buts illocutoires en correspondance avec les quatre directions d'ajustement entre les mots et les choses :

- **la direction d'ajustement des mots aux choses** : quand un acte illocutoire a cette direction d'ajustement, il est satisfait dans un contexte si son contenu propositionnel est vrai dans le monde au moment de l'énonciation. Le but des actes assertifs est de représenter comment les choses sont dans le monde ;
- **la direction d'ajustement des choses aux mots** : quand un acte illocutoire a cette direction d'ajustement, il est satisfait dans un contexte si les objets du monde ont été changés pour correspondre aux mots. Le but des actes directifs et commissifs ont la direction d'ajustement des mots aux choses. Ils ont pour but de transformer le monde, par l'action future du locuteur (actes commissifs) ou de l'interlocuteur (actes directifs) ;

- **la double direction des choses aux mots** : quand un acte illocutoire a cette direction d'ajustement, il est satisfait dans un contexte si le monde correspond au contenu propositionnel. Dans les actes déclaratifs, les objets de références sont transformés pour correspondre aux mots par le fait même de leur utilisation ;
- **la direction vide d'ajustement** : certains actes n'ont pas de véritable condition de satisfaction. Lors de l'énonciation d'actes expressifs, le locuteur exprime ses états d'âme sur les objets de références.

En résumé, la théorie des actes de langage est une théorie de la signification et de la communication qui analyse la forme logique des actes illocutoires et en propose une typologie. Les langages de communication d'agents s'inspirent de cette typologie. Ils mettent à disposition un répertoire d'actes de langage utilisables par des agents. Ces actes sont munis d'une sémantique intentionnelle.

1.3 Langage de communication d'agents

Pour communiquer, des agents doivent connaître, comprendre et utiliser un langage commun, appelé langage de communication d'agents (*ACL*, *i.e.* *Agent Communication Language*). À cet effet, un tel langage doit avoir :

- une **syntaxe** : on définit la structure des symboles ;
- une **sémantique** : on définit le sens de ces symboles ;
- une **pragmatique** : on définit la manière dont ils devront être utilisés.

Cette trichotomie fut l'objet de nombreuses critiques dans le domaine de la linguistique computationnelle. Étant donné que ces trois aspects ne sont pas aisément distinguables dans le langage naturel, une telle approche n'est pas viable pour l'analyse de texte. Toutefois, ce triptyque constitue un modèle adapté à l'élaboration d'un langage de communication d'agents. La définition de différents niveaux d'abstraction permet à une couche de s'affranchir des problèmes adressés par les couches de niveaux inférieurs.

Nous allons présenter dans cette section l'ACL le plus largement répandu qui est considéré comme un standard de fait : FIPA-ACL. Ce langage a été proposé par la FIPA⁴ (*Foundation for Intelligent Physical Agents*), organisation à but non lucratif, créée en 1996 et qui propose des standards pour l'interopérabilité des agents logiciels.

Dans la section 1.3.1, nous aborderons brièvement la syntaxe du langage FIPA-ACL. Nous présenterons ensuite sa sémantique (cf section 1.3.2). La section 1.3.3 est consacrée à la pragmatique des messages qui est au cœur de la problématique abordée par les langages de communication d'agents.

⁴<http://www.fipa.org>

1.3.1 Syntaxe

La syntaxe des messages FIPA-ACL est définie dans le document [7] sous la forme d'une grammaire BNF. Ce document propose également d'utiliser une grammaire DTD ⁵ pour XML⁶.

La figure 1.3.1.I présente une partie de cette grammaire. Par souci de concision, la grammaire DTD ne présente ici que 4 des 69 performatifs valides : `accept-proposal`, `agree`, `cancel` et `cfp`. Cette grammaire permet de plus l'adjonction de nouveaux paramètres par l'utilisateur (`user-defined`).

```

<!-- Document Type: XML DTD -->
<!-- Quatre des 69 performatifs valides-->
<!ENTITY %communicative-acts "accept-proposal
                                | agree
                                | cancel
                                | cfp ">
<!-- Les paramètres du messages -->
<!ENTITY %msg-param "receiver
                    | sender
                    | content
                    | language
                    | encoding
                    | ontology
                    | protocol
                    | reply-with
                    | in-reply-to
                    | reply-by
                    | reply-to
                    | conversation-id
                    | user-defined">
<!ELEMENT fipa-message ( %msg-param; )*>

```

Fig. 1.3.1.I: Grammaire DTD d'un message FIPA-ACL

⁵Document Type definition

⁶eXtensible Markup Language

1.3.2 Sémantique

La sémantique d'un message FIPA-ACL est définie dans [8]. Un message FIPA-ACL est constitué d'une liste non-ordonnée de 14 champs dont nous précisons ici le sens.

Un message FIPA-ACL comporte 2 champs (cf figure 1.3.1.I) relatifs à la communication. **sender** et **receiver** correspondent respectivement à l'agent émetteur et à l'agent récepteur du message.

Un message FIPA-ACL comporte 4 champs relatifs au contenu. Le champ **content** indique le contenu propositionnel du message. Le langage d'encodage du contenu est spécifié par le champ **encoding**. Ce contenu doit être une expression bien formée du langage de représentation spécifié par le champ **language**. Les constantes du contenu propositionnel doivent être définies dans une ontologie spécifiée par le champ **ontology**.

L'enchaînement des messages, appelé conversation, est géré à l'aide de cinq champs : **protocol** fait référence à l'un des onze protocoles proposés (par exemple, le protocole *Contract Net* [13]); **conversation-id** comporte l'identifiant de la conversation à laquelle appartient ce message; **reply-with**, comprend l'identifiant de ce message auquel il pourra être fait référence par la suite; **in-reply-to** englobe l'identifiant du message auquel celui-ci répond et **reply-by** renferme la date butoire de réponse. La notion de protocole sera abordée dans la section 1.4.

1.3.3 Pragmatique

La pragmatique des messages FIPA-ACL, *i.e.* la sémantique des performatifs est formulée en terme d'états mentaux. On parle alors d'**approche mentalistique**. La sémantique de ces performatifs est spécifiée non seulement en langage naturel mais également de manière formelle en SL (*Semantic Language*). La FIPA propose dans [12] une bibliothèque de 22 performatifs.

SL est un langage de logique modale du premier ordre muni de trois opérateurs (B, U, C). Les formules utilisant ces opérateurs modaux peuvent être interprétées de la manière suivante :

- $B_i p$: l'agent i croit p ;
- $C_i p$: l'agent i a pour but d'atteindre l'état du monde p ;
- $U_i p$: l'agent i doute de p mais l'estime plus vraisemblable que $\neg p$;

On définit de la manière suivante deux abréviations :

- $Bif_i p \equiv B_i p \vee B_i \neg p$: l'agent i croit p ou $\neg p$;
- $Uif_i p \equiv U_i p \vee U_i \neg p$: l'agent i doute de p ou de $\neg p$.

Afin de raisonner en terme d'actions, le langage est muni de termes pour représenter l'événement vide (e), une action (par exemple, a_i), une séquence d'actions (par exemple $a_i; a_j$), ou un choix non déterministe entre deux actions (par exemple $a_i|a_j$). Dans ce même but, quatre opérateurs ont été introduits (*Feasible*, *Done*, *Agent*, *Single*). Les formules utilisant ces opérateurs peuvent être interprétées de la manière suivante :

- $\text{Feasible}(a, p)$: l'action a est réalisable et a pour conséquence p ;
- $\text{Done}(a, p)$: l'action a a été effectuée et a eu pour conséquence p ;
- $\text{Agent}(i, a)$: l'agent i est l'unique agent qui effectue l'action a dans le passé et dans le futur ;
- $\text{Single}(a)$: l'action a est atomique. En d'autres termes, ce n'est pas une séquence d'action (on a $\text{Single}(a_i|a_j) \Leftrightarrow \text{Single}(a_i) \wedge \text{Single}(a_j)$) ;

C'est à partir des opérateurs précédents que les notions de but persistant et d'intention sont définies. On parle d'attitudes mentales composites. Ces attitudes mentales sont indispensables pour formaliser l'intention exprimée par l'illocution (cf section 1.2). La persistance d'un but peut, par exemple, être formalisée de la manière suivante :

Exemple n° 1.3.3.I [Persistance d'un but] *L'agent i a pour but persistant d'atteindre l'état du monde p si et seulement si :*

1. *il a pour but que p devienne vrai à un instant dans le futur et p n'est pas actuellement vrai.*
2. *avant que l'agent abandonne son but, une des deux conditions suivantes doit être remplie :*
 - (a) *l'agent croit que son but est satisfait ;*
 - (b) *l'agent croit que le but ne sera jamais satisfait.*

L'axiomatisation ainsi que la sémantique formelle associée aux opérateurs du langage SL est définie dans [11].

Un message FIPA-ACL muni d'un performatif act et d'un contenu p émis par l'agent i à l'agent j est noté $\langle i, \text{act}(j, p) \rangle$. Un message FIPA-ACL est associé d'une part à une formule qui conditionne son illocution (FP , *i.e.* *Feasibility Preconditions*). Cette formule correspond aux conditions de succès de l'acte de langage (cf section 1.2). Un message FIPA-ACL est associé d'autre part aux effets attendus sur les états mentaux du récepteur (RE , *i.e.* *Rational Effect*). Ces effets correspondent à la valeur perlocutoire de l'acte (cf section 1.2).

Exemple n° 1.3.3.II [Message FIPA-ACL] *On peut à titre d'exemple spécifier l'acte *inform* de la manière suivante :*

$$\begin{aligned} &\langle i, \text{inform}(j, p) \rangle \\ FP &: B_i p \wedge \neg B_i (B_i f_j p \vee U_i f_j p) \\ RE &: B_j p \end{aligned}$$

L'émetteur informe le récepteur de la véracité d'une proposition avec l'intention que ce dernier adopte cette croyance. Si l'agent sait au préalable que l'état du monde est atteint, en l'occurrence que le récepteur croit (ou se doute de) p (ou de $\neg p$), il ne peut pas avoir pour intention d'atteindre cet état du monde. À l'inverse, l'agent recevant un tel message sait que l'émetteur croit en p . De plus, il peut induire que le récepteur a pour intention de l'informer.

Les messages constituent des briques élémentaires qui doivent être ordonnées dans une conversation. Les agents doivent reconnaître les intentions des autres et raisonner à partir d'elles. L'implémentation de cette théorie pose le problème de l'automatisation de l'inférence [11]. Paradoxalement, contraindre la séquence des actions permet de réduire la complexité computationnelle liée à la génération d'une réponse [17]. C'est l'objet des protocoles d'interaction.

1.4 Protocole d'interaction

Une **conversation** est une séquence de messages échangés entre deux ou plusieurs agents. Elle doit être conforme à un protocole d'interaction.

|| Définition n° 1.4.0.I [Protocole d'interaction]

|| On appelle **protocole d'interaction** un ensemble de règles plus ou moins génériques qui sont partagées par les agents du système et auxquelles les interactions doivent se conformer.

Les protocoles permettent de guider l'interaction des agents. Ce sont des plans définis *a priori* pour coordonner leurs interactions. On affecte un **nom** à chacun des protocoles pour les identifier. Un certain nombre d'agents, deux ou plus, participent à une conversation. Le protocole spécifie le **rôle** qui leur est affecté. Par exemple, l'agent qui débute une conversation joue le rôle de l'initiateur. Le **comportement** d'un protocole décrit la séquence des messages qui peuvent être échangés. Ce comportement est spécifié dans un langage formel ou semi-formel.

Nous allons dans cette section explorer plusieurs formalismes pour spécifier le comportement d'un protocole et proposer un exemple pour chacun de ces formalismes. Nous débuterons par un langage de description semi-formelle : l'AUML (cf section 1.4.1). Nous envisagerons ensuite d'utiliser les automates finis déterministes pour décrire de manière plus formelle le comportement d'un protocole (cf section 1.4.2).

1.4.1 Langage semi-formel

Le langage de modélisation unifié UML⁷ (*Unified Modelling Language*) est une notation universelle pour la modélisation d'objets. Ce langage est très répandu dans le domaine du génie logiciel (*Software Engineering*). Toutefois, il ne permet pas de modéliser un système multi-agents. C'est dans cette perspective que le langage de modélisation unifié à base d'agents AUML⁸ (*Agent-based Unified Modelling Language*) a été proposé. Ce langage permet notamment de représenter le comportement des protocoles FIPA [14].

1.4.1.1 Formalisme

Le langage AUML étend la notion de diagramme de séquence utilisée dans UML 2.0 à celle de diagramme d'interaction qui permet de représenter les protocoles d'interaction et leurs instances, *i.e.* les conversations [29].

Un **diagramme d'interaction** est un graphique constitué d'un cadre qui permet d'identifier le protocole (nom, paramètres, etc. . .). Ce cadre délimite la représentation graphique du flux de messages entre les différents rôles. Un diagramme d'interaction est une représentation graphique en deux dimensions. Le temps est représenté selon un axe vertical. Les différents rôles joués par les participants sont représentés selon un axe horizontal.

En UML 2.0, une ligne de vie dans un diagramme de séquence est associée à une instance unique. En AUML, une ligne de vie dans un diagramme d'interaction est associée à un rôle. Une ligne de vie est représentée par un trait vertical pointillé surmonté d'un rectangle qui contient le nom du rôle associé.

Les messages FIPA-ACL (cf section 1.3) sont représentés dans un diagramme d'interaction à l'aide d'une flèche qui lie la ligne de vie du rôle joué par l'agent émetteur et la (ou les) ligne(s) de vie du (ou des) rôle(s) joué(s) par les agents destinataires. Dans le diagramme d'interaction d'un protocole, les messages ne sont pas totalement spécifiés. Seule une partie des champs est informée. En d'autres termes, les messages représentés dans un diagramme d'interaction sont plus ou moins génériques. Dans la plupart des cas, les agents sont libres d'en choisir le contenu (:**content**). Étant donné que les lignes de vie sont associées à des rôles, la cardinalité des messages doit être précisée. La cardinalité est une expression qui indique le nombre d'agents qui émettent et/ou reçoivent le type de message envisagé.

Les agents sont libres de choisir parmi une liste de messages génériques autorisés. Ce choix peut être représenté de manière compacte et concise à l'aide d'opérateur. Par

⁷<http://www.uml.org>

⁸<http://www.auml.org>

exemple, l'opérateur `or` dont le symbole est \diamond permet de sélectionner au moins un des messages génériques proposés.

La FIPA propose une librairie de protocole d'interaction [14]. Le protocole d'interaction « *Contract Net* » [13] fait partie de cette librairie.

1.4.1.2 Exemple

Reid G. Smith [61] est l'un des premiers chercheurs à envisager une approche multi-agents pour l'allocation de tâches. À cette intention, il utilise une métaphore organisationnelle. Les agents se coordonnent par l'intermédiaire de contrats qui sont négociés au travers d'un protocole intitulé « *Contract Net* ». Ce travail fut largement repris par la communauté SMA [20].

Un agent participant à une conversation conforme au protocole « *Contract Net* » peut jouer un des deux rôles suivants : soit il propose de sous-traiter une tâche en jouant le rôle de gestionnaire, soit il propose d'exécuter une tâche en jouant le rôle de contractant. Comment se déroule ce protocole ? Le gestionnaire commence par soumettre à un ensemble d'agents, *i.e.* les contractants, un appel à sous-traitance. Chacun des contractants évalue cette annonce et, lorsqu'il dispose des ressources nécessaires, il envoie au gestionnaire une proposition. Le gestionnaire rassemble toutes les propositions qu'il a reçues et alloue la tâche aux contractants qui ont fait les meilleures offres. Les contractants ainsi sélectionnés informent le gestionnaire du bon (ou du mauvais) déroulement de l'exécution.

Le protocole « *Contract Net* » [13] fait partie de la librairie des protocoles d'interaction proposés par la FIPA [14]. Son comportement est représenté dans la figure 1.4.1.I sous la forme d'un diagramme d'interaction. L'initiateur sollicite m participants lors d'un appel d'offre (`call for proposal`). Cet appel précise la tâche à effectuer ainsi que les conditions fixées par l'initiateur pour cette tâche. Il peut s'agir de contraintes sur le prix, le délai, ou plus généralement sur la qualité requise de la tâche. Les m participants qui reçoivent cet appel sont considérés comme des contractants potentiels. Parmi eux, n (avec $n \leq m$) participants réussissent à répondre à cet appel avant la date butoire. Ils émettent n réponses parmi lesquelles i (noté i^2n) sont négatives et j (avec $j \leq n$) sont positives (`propose`). Ces réponses précisent les conditions sous lesquelles la tâche peut être effectuée. Les $n - j$ autres réponses sont négatives (`refuse`). La date butoire de l'appel ayant été dépassée, l'initiateur est en mesure d'évaluer ces j propositions pour en sélectionner l (avec $l \leq j$). Il confirme ces délégations de tâches (`accept-proposal`). Les $j - l$ autres propositions sont refusées (`refuse-proposal`). Les l contractants ainsi sélectionnés informent le gestionnaire du bon (`inform-done`) ou du mauvais déroulement (`failure`) de la tâche voire du résultat (`inform-result`).

1.4.1.3 Limites

Le comportement d'un protocole est représenté graphiquement à l'aide d'un diagramme d'interaction. Ce diagramme est lisible et facile à comprendre pour un ingénieur logiciel familier avec le langage de modélisation unifié UML 2.0. De plus, un grand nombre d'outils permettent de générer automatiquement du code à partir de diagrammes UML. On peut citer le plugin eclipseUML développé par Omondo⁹. Cette même démarche peut être envisagée entre le code d'un agent logiciel et un diagramme AUML. Cependant, AUML n'est qu'un langage semi-formel. Il possède une syntaxe bien définie basée sur UML. La sémantique de ses notations restent néanmoins ambiguë car elles sont décrites en langage naturel. Les propriétés attendues du système sont généralement décrites de manière informelle et elles ne sont pas vérifiables. C'est la raison pour laquelle notre attention se tourne vers des langages plus formels.

1.4.2 Langage formel

Les automates sont des objets informatiques bien connus en théorie des langages. Un automate permet de décrire de manière formelle le comportement d'un protocole. Nous considérons ici l'exemple d'un protocole de marchandage.

1.4.2.1 Formalisme

Dans la théorie des langages, un automate est une procédure qui permet d'identifier si un mot appartient ou non à un langage régulier. De manière similaire, nous utiliserons ce formalisme pour décider si une conversation est conforme (ou non) à un protocole.

Les automates sont définis à partir d'un alphabet fini, c'est-à-dire un ensemble fini de caractères noté $\Sigma = \{a, b, c, \dots\}$. Nous désignerons par les lettres u, v, w, \dots les chaînes de caractères issues de cet alphabet, c'est-à-dire les éléments du monoïde Σ^* . λ désigne la chaîne vide. Un langage \mathcal{L} est un sous-ensemble quelconque de Σ^* . Les chaînes de ce langage sont appelées des mots. Un automate est une machine à état fini. On parle également de graphe d'état transition.

⁹<http://www.omondo.com/>

Définition n° 1.4.2.I [Automate fini]

Un **automate fini** est un quintuplet $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ où :

- Q est un ensemble fini d'états ;
- Σ est un alphabet fini ;
- $\delta : Q \times \Sigma \rightarrow 2^Q$ est une fonction de transition ;
- $q_0 \in Q$ est l'état initial ;
- $F \subseteq Q$ est un ensemble d'états finaux.

Dans ce formalisme graphique, les nœuds du graphe représentent des états et les arcs représentent des transitions d'un état à un autre.

On parle d'automate déterministe dans la mesure où à partir d'un état et d'une lettre, la fonction de transition permet d'atteindre au plus un état.

Définition n° 1.4.2.II [Automate fini déterministe]

Soit $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un automate fini. \mathcal{A} est un **Automate Fini Déterministe** (AFD) ssi :

$$\forall q \in Q \forall a \in \Sigma \|\delta(q, a)\| \leq 1$$

Dans la théorie des langages, un automate constitue une procédure qui permet de décider si un mot appartient (ou non) à un langage. On dit d'un tel langage qu'il est reconnu par un automate.

Définition n° 1.4.2.III [Langage reconnu]

Soit $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un automate fini. Soit $u = x_1.x_2 \dots x_n$ une chaîne de caractères issue de Σ . La chaîne de caractères est **acceptée par l'automate** \mathcal{A} ssi la séquence des états d'acceptation (q_0, \dots, q_n) est une séquence de $n + 1$ états de l'automate définie tq $\forall 0 \leq i \leq n - 1 \ q_{i+1} = \delta(q_i, x_{i+1})$ et $q_n \in F$. Un langage est **reconnu par un automate** ssi tous les mots de ce langage sont acceptés par cet automate.

D'après le théorème de Kleene, un langage \mathcal{L} est accepté par un automate \mathcal{A} ssi il constitue un ensemble régulier, c'est-à-dire s'il peut être défini par une expression régulière [62].

Exemple n° 1.4.2.I [Automate fini] Considérons, par exemple, l'automate \mathcal{A}_1 représenté dans la figure 1.4.2.1. Il comporte 5 états, $Q = \{1, 2, 3, 4, 5\}$. Il est défini sur un alphabet à six lettres $\Sigma = \{a, b, c, d, e, f\}$. L'état initial q_0 est l'état 1 représenté par un double cercle. Les états 4 et 5 sont finaux, $F = \{4, 5\}$. Ils sont représentés par des cercles noirs. La fonction de transition est représentée par les arcs reliant les états. Il

s'agit ici d'un automate fini déterministe puisque tous les arcs provenant d'un même état sont étiquetés par des lettres différentes. La chaîne de caractères $abac$ est acceptée par l'automate \mathcal{A}_1 . Le langage reconnu par cet automate est défini par l'expression régulière suivante :

$$[(b|\lambda)a(ba)^*(c|f)]|[(a|\lambda)b(ab)^*(d|e)]$$

Le comportement d'un protocole peut être représenté à l'aide d'un AFD. Les transitions sont alors étiquetées par des types de messages. On précise l'émetteur, le récepteur et le performatif utilisés. Les nœuds y représentent alors les états conversationnels d'un participant : soit un état d'émission soit un état de réception.

Définition n° 1.4.2.IV [Automate de comportement]

Soit $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un automate fini déterministe. \mathcal{A} représente le comportement d'un protocole qui met en jeu deux participants a et b ssi :

- Q est un ensemble fini d'états conversationnels qui sont soit des états d'émission pour l'agent a soit des états d'émission pour l'agent b ;
- Σ est un ensemble de messages génériques $\text{perf}(s, r)$, où perf est un performatif et s, r sont respectivement l'émetteur et le récepteur du message ;
- $\delta : Q \times \Sigma \rightarrow 2^Q$ est une fonction de transition ;
- $q_0 \in Q$ est l'état initial du protocole ;
- $F \subseteq Q$ est un ensemble d'états finaux du protocole.

Le langage reconnu par un tel automate peut être défini par une expression régulière dont les lettres représentent des types de messages. Les conversations doivent respecter cette expression régulière pour être conforme au protocole.

Ce formalisme permet de spécifier et de vérifier des propriétés formelles sur un protocole.

Définition n° 1.4.2.V [Protocole borné et vivace]

Soit $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ un automate fini déterministe qui représente le comportement d'un protocole. Ce protocole est :

- **vivace** ssi tous les états conversationnels sont accessibles depuis l'état initial ;
- **borné** ssi le langage reconnu est fini ;

Si un protocole est borné alors toutes les conversations qui s'y conforment sont finies. Si un protocole est vivace, tous les types de messages peuvent potentiellement être émis dans les conversations qui s'y conforment.

1.4.2.2 Exemple

Un protocole de marchandage met en jeu deux participants qui échangent des propositions et des contre-propositions dans le but de conclure une affaire, c'est-à-dire de trouver un compromis attractif pour les participants. C'est un échange de concessions.

Peynan Faratin propose dans sa thèse [22] de représenter le comportement de ce protocole à l'aide d'un automate. La figure 1.4.2.II présente une version simplifiée de cet automate. Les deux participants sont nommés a et b . $\text{proposal}(a, b)$ signifie que l'agent a fait une proposition à l'agent b . Parce que l'état 1 est initial, le protocole préconise de débiter par une proposition qui émane de l'un des participants. Parce que les états 4 et 5 sont finaux, la conversation se termine sur une acceptation (accept) ou sur un abandon (withdraw). La discussion consiste en un échange de propositions et de contre-propositions.

Pour être conforme au protocole, les conversations doivent respecter l'expression régulière suivante :

$$\begin{aligned} & [(\text{propose}(b, a)|\lambda)\text{propose}(a, b)(\text{propose}(b, a).\text{propose}(a, b))^* \\ & (\text{withdraw}(b, a)|\text{accept}(b, a))] \mid \\ & [(\text{propose}(a, b)|\lambda)\text{propose}(b, a)(\text{propose}(a, b).\text{propose}(b, a))^* \\ & (\text{withdraw}(a, b)|\text{accept}(a, b))] \end{aligned}$$

Ce langage n'est pas fini. En conséquence, le protocole de marchandage tel que nous l'avons formalisé n'est pas borné. De plus, ce protocole est vivace. En effet, tous les états conversationnels sont accessibles depuis l'état initial.

Un agent doit d'une part raisonner afin de vérifier que les conditions d'illocution (*Feasible Preconditions*) d'un message sont vérifiées. D'autre part, la stratégie d'un agent permet de sélectionner le message qui sera effectivement émis. En conséquence, on doit distinguer les propriétés du protocole et les propriétés des conversations conformes à ce protocole dans un SMA donné. Par exemple, le protocole de marchandage tel qu'il est formalisé ci-dessus n'est pas borné. Cela ne signifie pas pour autant que les conversations qui s'y conforment bouclent systématiquement. On peut définir la stratégie des participants pour qu'elle supprime la présence de cycle dans les conversations. De même, le protocole de marchandage tel qu'il est formalisé ci-dessus est vivace. Cela ne signifie pas pour autant que tous les performatifs seront effectivement utilisés dans les conversations. Là encore, on peut définir une stratégie qui permet d'atteindre systématiquement un consensus. Ainsi, le performatif withdraw ne sera jamais utilisé.

1.4.2.3 Limites

Utiliser un automate fini déterministe pour décrire le comportement d'un protocole permet de disposer d'une représentation graphique très lisible, compréhensible et intuitive. De plus, un tel formalisme est sans ambiguïté. Il permet de décrire le motif que doivent respecter les conversations qui se conforment au protocole. On peut donc envisager de générer automatiquement, à partir d'un automate de comportement, du code qui sera directement interprétable par un agent logiciel. Toutefois, ce formalisme nous limite aux protocoles qui impliquent deux participants.

Bien d'autres formalismes peuvent être utilisés pour spécifier le comportement d'un protocole : les réseaux de Petri [36], les réseaux de Petri colorés [59], le langage LOTOS [35] ou un langage de logique temporelle [35]. Quel que soit le formalisme utilisé, un certain nombre de propriétés sur les protocoles peuvent être spécifiées de manière formelle puis vérifiées.

1.5 Synthèse

Dans ce chapitre, nous avons montré que pour interagir des agents logiciels doivent partager un langage commun, appelé **langage de communication d'agents** (ACL).

La **théorie des actes de langage** permet d'identifier les constituants élémentaires de la communication. Selon cette théorie, la production d'un énoncé s'assimile à l'accomplissement d'une action : « *Parler, c'est agir!* » Proférer une énonciation est un acte intentionnel, appelé acte illocutoire. Cette théorie propose de classifier les verbes qui agissent, appelés performatifs, selon l'acte illocutoire qu'ils accomplissent. La plupart des langages de communication d'agents sont inspirés de la théorie des actes de langage.

C'est le cas du langage de communication d'agents FIPA-ACL. La pragmatique des messages, i.e. la sémantique des actes de langage est formulée en termes d'états mentaux. On parle alors d'**approche mentalistique**. Pour les trois raisons suivantes, cette approche ne permet pas de modéliser le dialogue au sens politique, c'est-à-dire comme une prise de décision concertée.

1. La théorie des actes de langage considère les énoncés comme des épiphénomènes. En pratique, les messages constituent des briques élémentaires qui doivent être ordonnées dans une conversation, conformément à un protocole d'interaction, i.e. une convention sociale qui régit l'enchaînement des messages. D'une part, ces protocoles sont trop rigides. Par exemple, les caractéristiques d'une tâche ne peuvent être modifiées au travers du protocole « Contract

Net », elles sont fixées par l'appel d'offre. D'autre part, parce qu'un protocole ne spécifie ni le raisonnement des agents ni leurs stratégies, ses propriétés ne nous permettent pas de garantir l'obtention d'un résultat au terme d'une conversation et de préjuger de sa qualité.

2. Considérée comme la clé pour l'usage et la compréhension d'un acte de langage par un agent, la valeur illocutoire doit faire l'objet d'une convention. Cette valeur institutionnelle n'est pas explicite dans la théorie des actes de langage. En conséquence, la sémantique des actes de langage est privée et subjective. Un agent qui assiste à la conversation ne confère pas le même sens aux actes de langage que le destinataire. *A fortiori*, il ne réalise pas les mêmes inférences.
3. La sémantique des actes de langage est spécifiée en termes d'états mentaux. Ce modèle de raisonnement n'est pas adapté à la gestion de conflits. Par exemple, lorsqu'un agent reçoit une information de la part d'un autre agent son état mental est modifié. L'information reçue est systématiquement adoptée qu'elle confirme ou infirme le point de vue de l'agent.

Pour pallier cette dernière lacune, nous allons présenter dans le chapitre 2 un modèle de raisonnement qui permet à un agent de gérer les incohérences entre ses propres croyances et les informations transmises par d'autres agents.

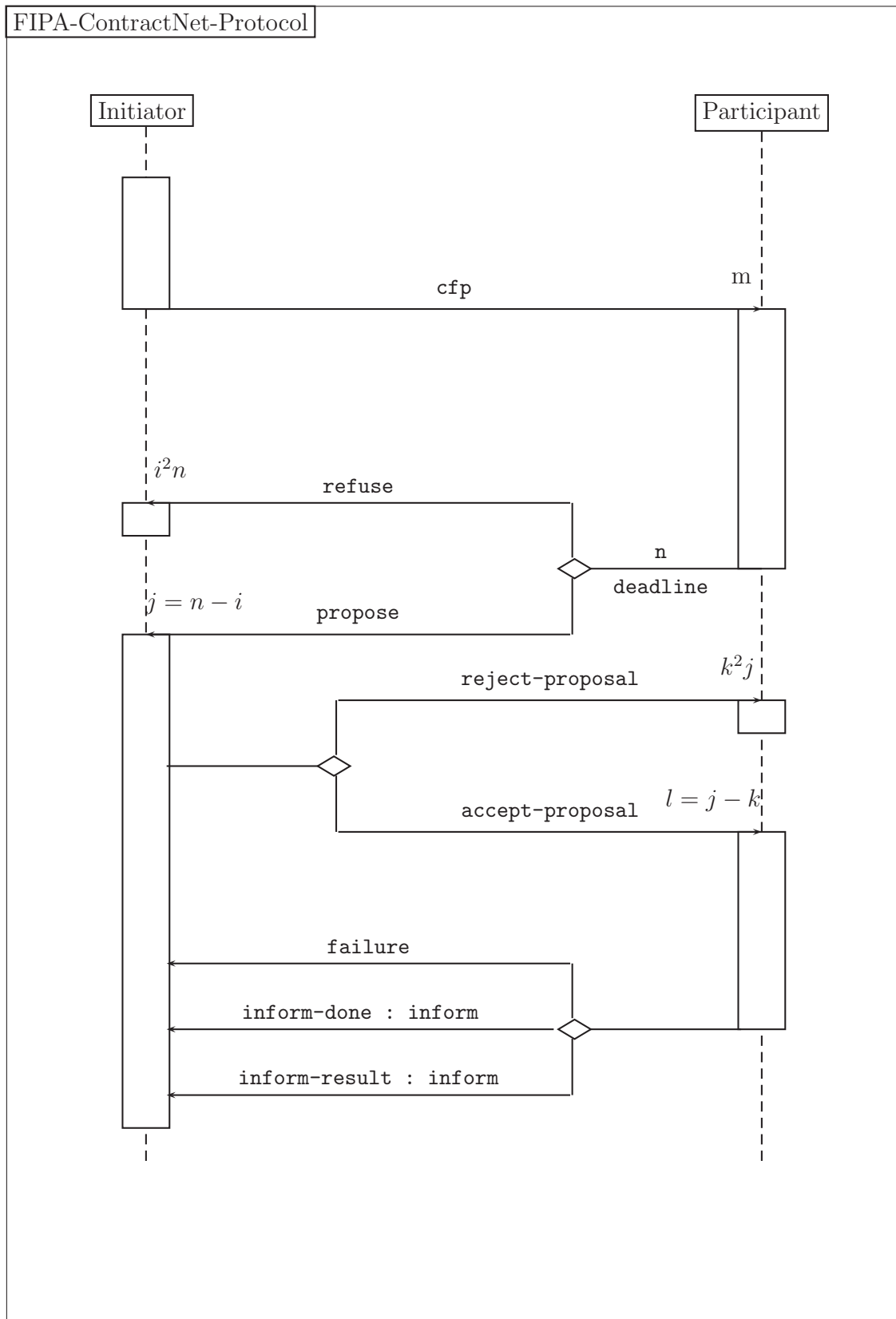


Fig. 1.4.1.I: Protocole FIPA « Contract-Net »

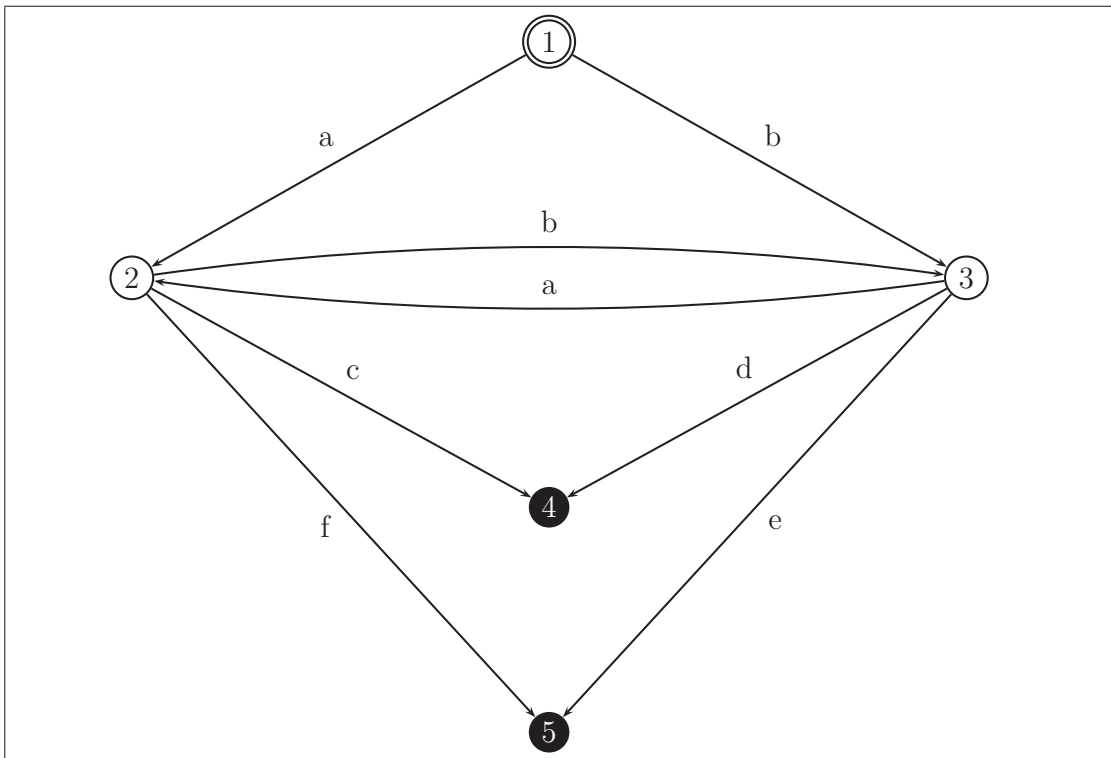


Fig. 1.4.2.I: Représentation d'un automate

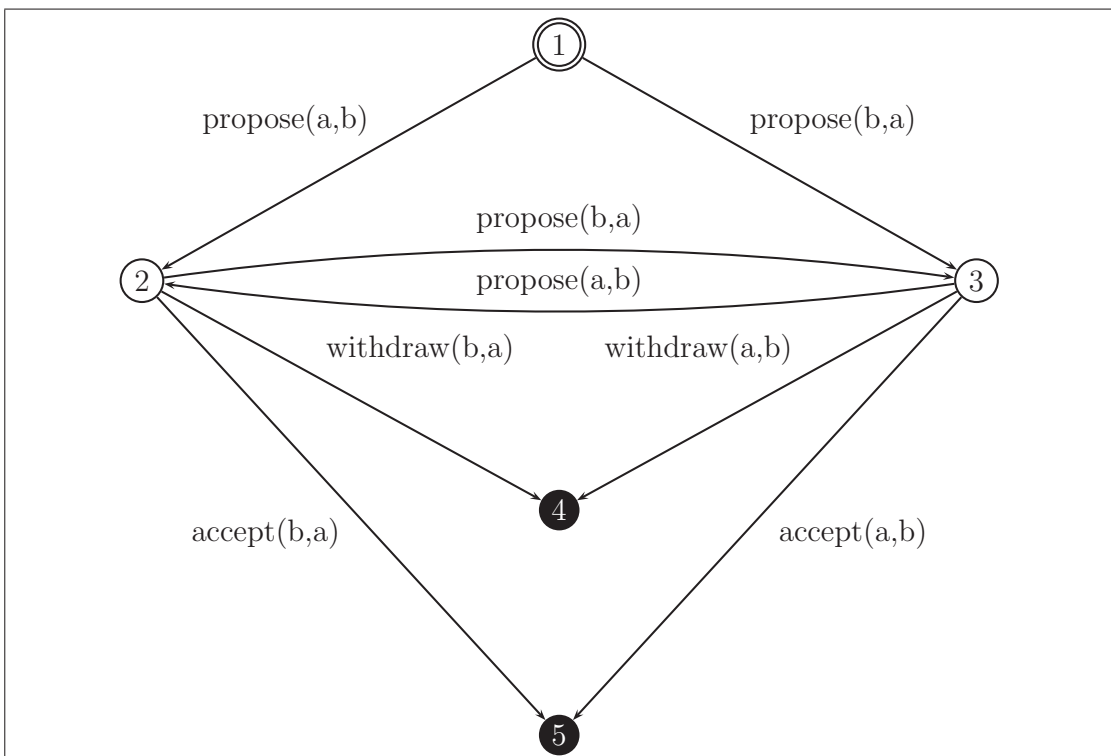


Fig. 1.4.2.II: Protocole de marchandage

“The one who has the last word
laughs best.”

Adage

Chapitre 2

Argumentation

Sommaire

2.1	Introduction	29
2.2	Argument	30
2.2.1	Motivation	30
2.2.2	Structure des arguments	31
2.2.3	Force des arguments	32
2.3	Systèmes d’argumentation abstraits	33
2.3.1	Système d’argumentation	34
2.3.2	Système d’argumentation à base de préférences	39
2.3.3	Système d’argumentation à base de valeurs	41
2.4	Logiques argumentatives	44
2.4.1	Relation de contradiction	45
2.4.2	Relation de priorité statique	48
2.4.3	Relation de priorité dynamique	52
2.5	Synthèse	57

2.1 Introduction

Lorsqu’un agent social et autonome raisonne et interagit avec le monde qui l’entoure, il peut être confronté à différentes sources d’incohérence : croyances erronées, observations non fiables, échanges d’informations avec d’autres agents, etc . . . Un agent intelligent doit donc disposer d’un mécanisme de raisonnement qui permet de gérer ces incohérences. L’argumentation est un bon candidat. L’argumentation constitue un modèle adapté au processus cognitif d’un agent autonome et social pour gérer les interactions entre des arguments internes qui explicitent ses croyances et des

arguments externes qui créditent des croyances contradictoires provenant d'autres agents.

Dans un premier temps, nous allons dans la section 2.2 présenter la notion d'argument et ses fondements philosophiques. Dans un second temps, nous allons présenter dans quelle mesure ces travaux ont influencé les modèles d'argumentation en IA. D'une part, on considère dans un système d'argumentation les arguments comme des entités abstraites (cf section 2.3). D'autre part, on considère dans une logique argumentative que les arguments sont munis d'une structure (cf section 2.4).

2.2 Argument

L'argumentation permet de tirer des conclusions en se réservant le droit de les rétracter à la lumière de nouvelles informations (cf section 2.2.1). D'une part, Stephen Toulmin fonde l'argumentation sur la notion d'évidence afin de donner une structure aux arguments (cf section 2.2.2). D'autre part, Chaïm Perelman et Lucie Olbrechts-Tyteca fondent l'argumentation sur la notion d'adhésion afin de donner une force aux arguments (cf section 2.2.3).

2.2.1 Motivation

La déduction classique (cf Annexe A.4) est une opération de conséquence (notée \vdash) qui permet de calculer la signification des formules en construisant des preuves. Par exemple, la preuve présentée ci-dessous permet, à partir d'un certain nombre de prémisses, de conclure qu'Olivier est vieux :

Olivier a 84 ans.

Olivier est un homme.

Tous les hommes de plus de 80 ans sont vieux.

$84 \geq 80$.

Olivier est vieux.

La déduction classique vérifie la propriété de monotonie : si Φ est une conséquence de Γ alors c'est également une conséquence de tout ensemble qui contient Γ :

$$\text{si } \Gamma \vdash \Phi \text{ et } \Gamma \subset \Delta \text{ alors } \Delta \vdash \Phi.$$

En d'autres termes, l'adjonction de nouvelles formules à un ensemble Γ ne peut jamais remettre en cause ce que l'on peut déduire de Γ . Une preuve nous contraint à en accepter la conclusion dans la mesure où on en accepte les prémisses. On parle de **monde clos**.

Contrairement à la logique classique, l'argumentation permet de tirer des conclusions en se réservant le droit de les rétracter à la lumière de nouvelles informations. On construit et on compare des arguments en faveur d'une conclusion ainsi que des contre-arguments en défaveur de cette même conclusion. Par exemple, l'argument suivant permet d'aboutir à la même conclusion que précédemment : « **Olivier est vieux parce qu'il est octogénaire.** »

Les prémisses d'un argument peuvent générer des contre-arguments qui le contredisent. Contrairement à une preuve, un argument peut être défait (*defeat*). Un argument est ouvert aux objections. On parle de **monde ouvert**.

Un argument n'est admis qu'à la condition que toutes les objections qui ont été émises sont à leur tour défaites. On distingue différentes sources d'objections :

1. les arguments laissent certaines prémisses implicites en supposant que l'audience y adhère. Par exemple, on présuppose qu'Olivier est un homme ;
2. les arguments utilisent des termes vagues, imprécis ou des formules ouvertes. Par exemple, aucun seuil d'âge n'est mentionné. On peut demander à le préciser ;
3. les arguments admettent des objections dans des cas exceptionnels. Par exemple, Olivier est peut-être immortel ;
4. les arguments peuvent être avancés même lorsque l'on doute de certains faits. Par exemple, je ne suis pas certain qu'Olivier soit octogénaire.

Cette liste n'a pas la prétention d'être exhaustive. Elle permet cependant d'établir une distinction entre un argument et une preuve.

En résumé, lorsqu'on dispose d'informations incomplètes, incertaines ou imprécises, lorsque le monde est ouvert, il est préférable d'utiliser un système d'argumentation pour modéliser le raisonnement plutôt qu'un système de preuves. C'est le cas d'un agent autonome et social. Un argument peut être muni d'une structure et d'une force.

2.2.2 Structure des arguments

Stephen Toulmin¹ définit l'argumentation grâce à la notion de vérité. Son ouvrage majeur intitulé "*The use of argument*" [57] a été une source d'inspiration majeure en logique et en IA. Ce travail se focalise sur la justification d'un argument lorsqu'il est calomnié (*challenge*).

¹Philosophe anglais, actuellement professeur à l'USC (1922-)

Stephen Toulmin envisage dans [57] l'argumentation comme la construction d'un schéma d'analyse vers une conclusion à l'aide de données et de justifications. Comme l'évoque la figure 2.2.2.I², il distingue six briques de base :

- la conclusion (*claim*), *i.e.* le fait que l'on va justifier ;
- les données (*data*) *i.e.* les faits sur lesquels l'argument s'appuie ;
- la garantie (*warrant*) légitime le saut entre les données et la conclusion ;
- le fondement (*backing*), *i.e.* le support à la garantie ;
- le qualificateur modal (*modal qualifier*) indique la force du saut entre les données et la garantie ;
- la condition d'exception (*rebutal*) apparaît lorsque le saut entre les données et la garantie n'est pas légitime.

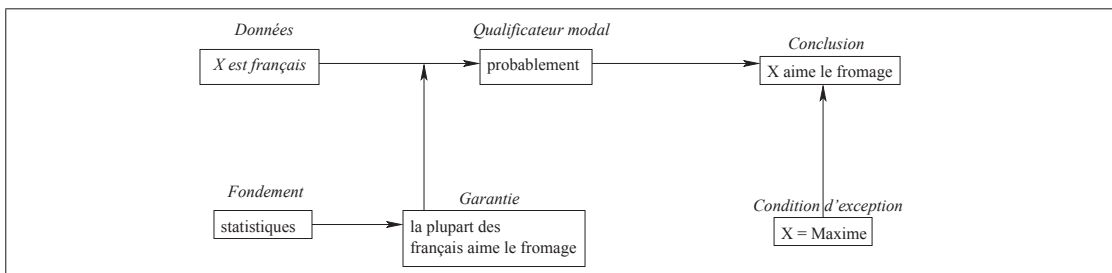


Fig. 2.2.2.I: Schéma de l'argumentation selon Stephen Toulmin

En résumé, un argument est un schéma qui permet d'aboutir à une conclusion. Les références à ce schéma sont multiples dans les travaux en logique argumentative (cf section 2.4). Le travail de Chaïm Perelman présenté ci-dessous s'appuie plus sur la notion d'adhésion que sur la notion d'évidence.

2.2.3 Force des arguments

Chaïm Perelman³ et sa collaboratrice Lucie Olbrechts-Tyteca publient en 1958 un ouvrage majeur intitulé "Traité de l'Argumentation" [15]. D'après cet ouvrage, la rhétorique ne doit pas seulement prendre en considération l'aspect factuel mais également l'importance que l'audience attache aux faits.

Ainsi, l'objet de la théorie de l'argumentation est : *... l'étude des techniques discursives permettant de provoquer ou d'accroître l'adhésion des esprits aux thèses qu'on présente à leur assentiment ...* Gaëlle Desbordes et Bernard Moulin proposent dans [18] une formalisation de ces techniques à l'aide du modèle agent et de graphes conceptuels.

²NdR : Maxime aime toutefois le gruyère, la mozzarella, la raclette, le maroille et le reblochon.

³Philosophe polonais (1912-1984) professeur de logique, de morale et de métaphysique à l'Université de Bruxelles jusqu'en 1978

La notion d'évidence est rejetée au profit de celle d'adhésion. À la différence de l'évidence, l'adhésion implique l'orateur, *i.e.* la personne qui argumente et surtout l'audience, *i.e.* la personne à laquelle s'adresse l'argumentation. L'adhésion des esprits est caractérisée par une intensité variable, pas nécessairement proportionnelle à la probabilité d'une thèse. Toute argumentation est relative à l'audience qu'elle cherche à influencer. Perelman et Olbrechts-Tyteca différencient deux types d'audience : l'audience universelle et l'audience formée.

D'une part, l'**audience universelle** est constituée par l'humanité tout entière. Une argumentation qui s'adresse à une audience universelle doit convaincre de l'évidence des raisons fournies, c'est-à-dire de leur validité absolue et intemporelle. La rhétorique efficace pour une audience universelle est celle qui manie la preuve logique. Le consensus repose sur des faits et des vérités.

D'autre part, l'**audience formée** est constituée de la seule personne à laquelle on s'adresse, *i.e.* l'interlocuteur dans le dialogue. Une argumentation qui s'adresse à une audience formée pose le problème de la controverse. Convaincre quelqu'un, c'est vaincre par des arguments plus forts. Le consensus repose sur des valeurs et sur leurs hiérarchies. Quand elles sont vagues, les valeurs sont universelles et quasi-semblables à des vérités. Quand elles sont précises, les valeurs se présentent simplement comme conformes aux aspirations de certains groupes particuliers.

En résumé, différentes personnes peuvent faire des choix différents selon leurs préférences, c'est-à-dire l'échelle de valeurs à laquelle elles se réfèrent. Un argument persuasif pour une audience ne l'est pas nécessairement pour une autre audience. Les préférences peuvent (ou non) être prises en compte dans les systèmes d'argumentation abstraits présentés dans la section suivante.

2.3 Systèmes d'argumentation abstraits

Nous allons tout d'abord présenter le modèle de calcul argumentatif défini par Phan Minh. Dung proposé dans [21]. Ce travail a largement inspiré la plupart des systèmes d'argumentation proposés à ce jour. Nous envisagerons ensuite deux extensions de ce cadre [1, 9] qui permettent d'affecter une force aux arguments. À cette intention, le système d'argumentation proposé par Leila Amgoud et Claudette Cayrol [1, 2] incorpore la notion de préférence (cf section 2.3.2) et le système d'argumentation proposé par Trevor Bench-Capon incorpore la notion de valeur (cf section 2.3.3).

2.3.1 Système d'argumentation

Phan Minh. Dung propose dans [21] un modèle de calcul pour modéliser l'argumentation humaine lors de la résolution de problèmes. Un argument est dit acceptable s'il peut être soutenu malgré les critiques qui lui sont attribuées. L'acceptabilité d'un argument dépend des arguments (respectivement des contre-arguments) qui le défendent (respectivement qui l'attaquent).

Ainsi, un argument est une entité abstraite dont la sémantique n'est définie que par les relations qui le lient avec les autres arguments. C'est la raison pour laquelle on parle de système d'argumentation abstrait. Par souci de concision, nous employerons ici le terme de système d'argumentation.

Définition n° 2.3.1.I [Système d'argumentation]

Un **système d'argumentation** est un couple $AS = \langle AR, attacks \rangle$ où :

- AR est un ensemble fini d'arguments ;
- $attacks$ une relation binaire sur AR ($attacks \subseteq AR \times AR$).

$attacks(P, Q)$ se lit « P attaque Q » ou « Q est attaqué par P ».

Par abus de langage, on dit qu'un ensemble d'arguments S attaque Q , ou bien Q est attaqué par S (noté $attacks(S, Q)$) ssi un argument de S attaque Q .

Exemple n° 2.3.1.I [Système d'argumentation] Dans la figure 2.3.1.1, le système d'argumentation AS est représenté à l'aide d'un graphe orienté dont les sommets sont des arguments et les arcs représentent des relations d'attaque.

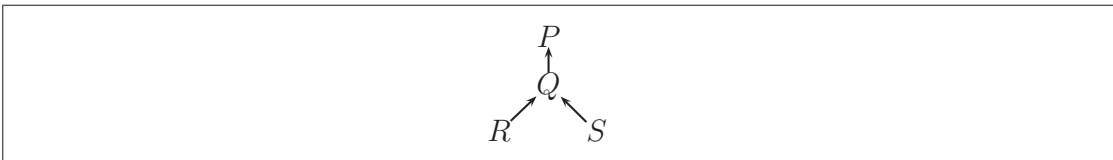


Fig. 2.3.1.I: Représentation sous forme de graphe orienté du système d'argumentation AS

L'objectif de ce système consiste à déterminer dans quelle mesure un argument est acceptable ou non. La sémantique d'un système d'argumentation assigne un statut aux arguments dans un ensemble donné en fonction des relations qui les lient les uns aux autres. Un argument est acceptable si les autres arguments acceptables le défendent des attaques qui lui sont portées. C'est dans cette perspective que se situent les définitions suivantes.

Définition n° 2.3.1.II [Libre de conflit/Acceptabilité/Admissibilité]

Soient $AS = \langle AR, attacks \rangle$ un système d'argumentation, $S \subseteq AR$ un ensemble d'arguments et $P \in AR$ un argument.

1. S est **libre de conflit** ssi $\forall P, Q \in S \neg attacks(P, Q)$;
2. P est **acceptable** vis à vis de S (noté $acc_{AS}^S(P)$) ssi $\forall Q \in AR attacks(Q, P) \Rightarrow attacks(S, Q)$;
3. S libre de conflit est **admissible** (noté $adm_{AS}(S)$) ssi $\forall Q \in S acc_{AS}^S(Q)$.

En logique classique, une théorie a une unique extension qui est un point fixe de l'opération de déduction. En logique non-monotone, une théorie peut admettre plusieurs extensions. L'argumentation constitue un modèle de raisonnement non-monotone dont la sémantique est définie par différentes extensions. À cet effet, on considère ici trois types d'extension : l'extension raisonnable (*ground extension*), l'extension préférée (*preferred extension*), l'extension stable (*stable extension*).

L'extension raisonnable est définie à l'aide de la fonction caractéristique du système d'argumentation.

Définition n° 2.3.1.III [Fonction caractéristique]

Soit $AS = \langle AR, attacks \rangle$ un système d'argumentation. La **fonction caractéristique** de ce système d'argumentation (notée F_{AS}) est définie telle que :

$$F_{AS} : 2^{AR} \rightarrow 2^{AR}$$

$$F_{AS}(S) = \{P \in AR ; acc_{AS}^S(P)\}$$

Il est aisé de s'apercevoir que si un argument est acceptable étant donné un ensemble d'arguments alors il est acceptable étant donné un de ses sur-ensembles. En d'autres termes, F_{AS} est monotone par rapport à l'inclusion ensembliste.

Définition n° 2.3.1.IV [Extension raisonnable]

Soit $AS = \langle AR, attacks \rangle$ un système d'argumentation. L'**extension raisonnable** de ce système d'argumentation est le plus petit point fixe de F_{AS} . On note $GE_{AS}(S)$ le prédicat correspondant.

En d'autres termes, une extension raisonnable peut être calculée par l'algorithme 2.3.1.

Exemple n° 2.3.1.II [Extension raisonnable] Dans la figure 2.3.1.III, les systèmes d'argumentation AS_1 et AS_2 sont représentés respectivement à gauche et à droite.

Entrée : AS
Sortie : S
 $S = \emptyset$;
répéter
 | $S' = S$;
 | $S = S \cup \{P \in AS; \text{acc}_{AS}^S(P)\}$
jusqu'à $S = S'$;

Fig. 2.3.1.II: Algorithme pour la construction d'une extension raisonnable d'un système d'argumentation

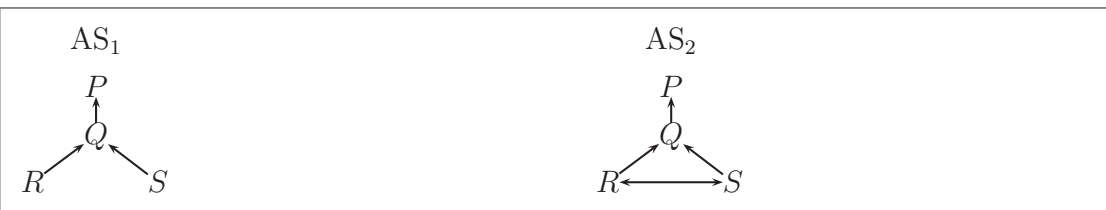


Fig. 2.3.1.III: Représentation sous forme de graphe orienté des systèmes d'argumentation AS₁ (à gauche) et AS₂ (à droite)

Chacun de ces systèmes d'argumentation possède une unique extension raisonnable : $GE_{AS_1}(\{R, S, P\})$ et $GE_{AS_2}(\emptyset)$.

Un système d'argumentation a toujours une unique extension raisonnable qui peut être éventuellement vide. Afin d'enrichir la sémantique d'un système d'argumentation, on définit une extension préférée comme un ensemble admissible maximal.

Définition n° 2.3.1.V [Extension préférée]

Soit $AS = \langle AR, attacks \rangle$ un système d'argumentation. Une **extension préférée** de ce système d'argumentation est un ensemble admissible d'arguments S qui est maximal (au sens de l'inclusion). On note $\text{pref}_{AS}(S)$ le prédicat correspondant.

Une extension préférée représente une position consistante qui peut se défendre par elle-même de toute attaque et qui ne peut pas être étendue sans introduire de conflit.

Exemple n° 2.3.1.III [Extension préférée] Le système d'argumentation AS₂ présenté dans l'exemple précédent possède une unique extension raisonnable : $GE_{AS_2}(\emptyset)$ et deux extensions préférées : $\text{pref}_{AS_2}(\{P, R\})$ et $\text{pref}_{AS_2}(\{P, S\})$.

Un système d'argumentation n'a pas nécessairement une unique extension préférée. On peut alors distinguer plusieurs classes d'acceptabilité.

|| **Définition n° 2.3.1.VI [Crédulité/Scepticisme]**

|| Une rationalité qui accepte seulement les arguments qui sont dans toutes les extensions préférées est dite **sceptique**. Si elle accepte les arguments qui sont dans au moins une extension préférée, la rationalité est qualifiée de **crédule**.

L'extension raisonnable est incluse dans toutes les extensions préférées. De plus, une extension préférée peut être éventuellement vide. C'est la raison pour laquelle l'extension stable a été introduite.

|| **Définition n° 2.3.1.VII [Extension stable]**

|| Soit $AS = \langle AR, attacks \rangle$ un système d'argumentation. Un ensemble d'arguments $S \subseteq AR$ libre de conflit est appelé **extension stable** ssi $\forall P \in AR - S$ attacks(S, P). On note le prédicat correspondant ainsi $stable_{AS}(S)$.

Une extension stable est un ensemble libre de conflit qui attaque tous les arguments externes.

Exemple n° 2.3.1.IV [Extension stable] Dans la figure 2.3.1.IV, sont représentés les systèmes d'argumentation AS_2 et AS_3 respectivement à gauche et à droite. Les deux

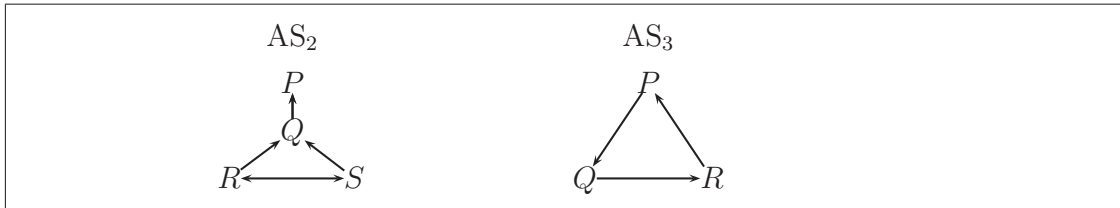


Fig. 2.3.1.IV: Représentation sous forme de graphe orienté des systèmes d'argumentation AS_2 (à gauche) et AS_3 (à droite)

extensions préférées ($pref_{AS_2}(\{P, R\})$ et $pref_{AS_2}(\{P, S\})$) de AS_2 sont également des *extensions stables*. Le système d'argumentation AS_3 ne possède pas d'extension stable.

Un système d'argumentation n'a pas nécessairement d'extension stable. Toutefois lorsqu'elle existe, elle n'est pas vide. De plus, toute extension stable est une extension préférée. La réciproque n'est pas nécessairement vraie. Si elle est vérifiée, on parle de système d'argumentation cohérent.

|| **Définition n° 2.3.1.VIII [Système d'argumentation cohérent]**

|| Soit $AS = \langle AR, attacks \rangle$ un système d'argumentation. AS est dit **cohérent** ssi toute extension préférée est également une extension stable.

Le tableau 2.3.1.I synthétise la complexité algorithmique d'un certain nombre de problèmes de décision [46]. Le lecteur peu familier avec les classes de complexité peut se référer à l'annexe B.

Problème	Description	Complexité
$\text{adm}_{\text{AS}}(S)$	Est-ce que S est un ensemble admissible ?	\mathcal{P}
$\text{GE}_{\text{AS}}(S)$	Est-ce que S est une extension raisonnable ?	\mathcal{P}
$\text{pref}_{\text{AS}}(S)$	Est-ce que S est une extension préférée ?	$\text{co}\mathcal{NP}$ -complet
$\text{stable}_{\text{AS}}(S)$	Est-ce que S est une extension stable ?	\mathcal{P}
$\text{has-stable}(\text{AS})$	Est-ce que AS possède une extension stable ?	\mathcal{NP} -complet
$\text{c}(P, \text{AS})$	Est-ce que P est accepté par un agent crédule ?	\mathcal{NP} -complet
$\text{s}(P, \text{AS})$	Est-ce que P est accepté par un agent sceptique ?	$\Pi_2\mathcal{P}$

Tab. 2.3.1.I: Complexité algorithmique des problèmes de décision liés au système d'argumentation

Contrairement à une extension préférée, une extension stable n'est jamais vide mais elle n'existe pas systématiquement. Identifier une extension stable est algorithmiquement plus facile qu'identifier une extension préférée. Toutefois déterminer son existence est \mathcal{NP} -complet .

La notion d'extension préférée est plus intéressante que celle d'extension raisonnable puisqu'elle permet d'introduire différentes interprétations : le scepticisme et la crédulité. Toutefois elle est algorithmiquement plus difficile à identifier sauf sous certaines conditions.

Théorème n° 2.3.1.I [Système d'argumentation sans circuit]

Soit $\text{AS} = \langle \text{AR}, \text{attacks} \rangle$ un système d'argumentation. Si le graphe associé ne contient pas de circuit alors ce système possède une unique extension préférée non-vide qui est également une extension raisonnable et une extension stable.

Par conséquent, le problème de décision $\text{pref}_{\text{AS}}(S)$ est sous ces conditions polynomial⁴.

Il est important de noter que la plupart des approches en raisonnement non-monotone et en programmation logique ne sont que des formes particulières de cette théorie. En effet, Dung [21] précise les relations qui existent entre les différentes extensions et celles des autres formes de raisonnements non-monotones. Cela explique en grande partie la popularité qu'a eue ce travail.

⁴Ça roule avec un système d'argumentation sans circuit !

Toutefois, dans le cadre proposé par Dung, une attaque d'un argument sur un autre argument est systématiquement victorieuse. Ce choix n'est pas toujours approprié. Par exemple, les principes de *lex posteriori* et de *lex specialis* permettent de trancher entre deux arguments conflictuels.

Exemple n° 2.3.1.V [lex posteriori/lex specialis] *Dans le domaine législatif, se pose la question de l'attitude du juge lorsqu'il existe des relations conflictuelles entre les sources du droit. Dès lors, le droit prévoit des règles qui permettent de circonscrire le conflit. On peut citer le principe de lex posteriori qui se fonde sur l'apparition dans le temps des lois : la loi la plus récente va s'imposer devant celle plus ancienne. On peut citer également le principe de lex specialis : la loi spécifique s'applique aux dépens de la loi générale.*

Il est parfois souhaitable de pouvoir choisir entre deux arguments qui s'attaquent mutuellement. Notre tempérament, nos préférences personnelles, le contexte peuvent nous amener à considérer un argument plus fort qu'un autre argument. On ignore alors l'une de ces attaques. La prise en compte des préférences entre les arguments constitue l'objectif du travail présenté dans la section suivante.

2.3.2 Système d'argumentation à base de préférences

Leila Amgoud et Claudette Cayrol étendent dans [2, 1] le modèle de calcul précédent. L'acceptabilité d'un argument dépend des arguments (respectivement des contre-arguments) qui le défendent (respectivement qui l'attaquent) mais également de la force de ces arguments.

Ainsi, le système d'argumentation (cf définition 2.3.1.I) est muni d'une relation de préférence qui lie les arguments.

Définition n° 2.3.2.I [Système d'argumentation à base de préférences]

Un **système d'argumentation à base de préférences** est un triplet $PAS = \langle AR, attacks, pref \rangle$ où :

- $\langle AR, attacks \rangle$ est un système d'argumentation tel que nous l'avons défini précédemment (cf définition 2.3.1.I) ;
- $pref$ une relation de préférences, c'est à dire une relation d'ordre strict sur AR ($pref \subset AR \times AR$).

$pref(P, Q)$ se lit « P est préféré à Q ».

Les arguments sont liés par une relation de préférence, c'est-à-dire une relation asymétrique et transitive (cf annexe C).

De cette manière, l'attaque d'un argument P contre un argument Q peut échouer si Q est préféré à P . C'est la raison pour laquelle la définition suivante a été introduite.

Définition n° 2.3.2.II [Défaire]

Soient $PAS = \langle AR, attacks, pref \rangle$ un système d'argumentation à base de préférences, $P, Q \in AR$ deux arguments. P **défait** Q (noté $defeats(P, Q)$) ssi on a $attacks(P, Q) \wedge \neg pref(Q, P)$.

Par abus de langage, on dit qu'un ensemble d'arguments S défait Q ou bien Q est défait par S (noté $defeats(S, Q)$) ssi un argument de S défait Q .

En d'autres termes, P attaque avec succès Q si Q n'est pas préféré à P . La définition 2.3.1.II a par conséquent été modifiée de la manière suivante :

Définition n° 2.3.2.III [Libre de conflit/Acceptabilité/Admissibilité]

Soient $PAS = \langle AR, attacks, pref \rangle$ un système d'argumentation à base de préférences, $S \subseteq AR$ un ensemble d'arguments et $P, Q \in AR$ deux arguments.

1. S est **libre de conflit** ssi $\forall P, Q \in S \neg defeats(P, Q)$;
2. P est **acceptable** vis à vis de S (noté $acc_{PAS}^S(P)$) ssi $\forall Q \in AR \text{ } defeats(Q, P) \Rightarrow defeats(S, Q)$;
3. S libre de conflit est **admissible** (noté $adm_{PAS}(S)$) ssi $\forall P \in S \text{ } acc_{PAS}^S(P)$.

Comme dans la définition 2.3.1.V, une extension préférée est un ensemble admissible d'arguments qui est maximal au sens de l'inclusion. On note $pref_{PAS}(S)$ le prédicat correspondant.

On peut représenter un système d'argumentation à base de préférences au moyen d'un graphe orienté. Les sommets sont des arguments qui sont reliés par des arcs si l'argument représenté par le nœud source de l'arc défait l'argument représenté par le nœud destination de l'arc. La relation de préférences étant une relation d'ordre strict, elle est asymétrique et transitive (cf annexe C). Par conséquence, un tel graphe ne possède pas de circuit. D'après le théorème 2.3.1.I, l'extension préférée correspond à l'extension raisonnable. Elle est unique et non-vide. De plus, le problème de décision associé est polynomial.

Exemple n° 2.3.2.I [Système d'argumentation à base de préférences] La figure 2.3.2.1 représente le système d'argumentation à base de préférence PAS_4 . À gauche, les relations d'attaque sont représentées à l'aide d'un graphe orienté. Au centre sont énumérées les relations de préférence. PAS_4 est représenté par le graphe orienté à droite. L'extension préférée ($pref_{AS_4}(\{P, R\})$) de AS_4 , qui est également une extension raisonnable, est non-vide et unique.

En résumé, l'extension du système d'argumentation présentée ici permet d'incorporer la notion de préférences. On peut donc modéliser les choix réalisés par ce

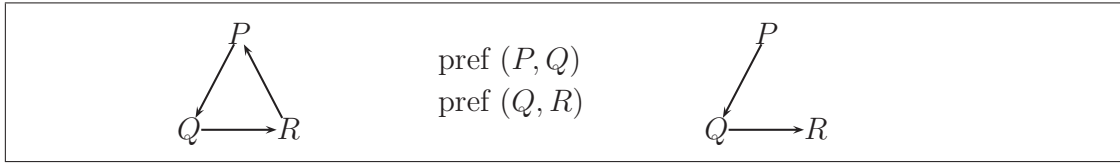


Fig. 2.3.2.I: Représentation du système d'argumentation à base de préférences AS_4

que Chaïm Perelman appelle une audience formée, *i.e.* une audience particulière de l'argumentation. Toutefois, cette approche ne nous permet pas de considérer simultanément plusieurs audiences. C'est l'objet des travaux présentés dans la section suivante.

2.3.3 Système d'argumentation à base de valeurs

Trevor Bench-Capon [9] étend lui aussi le système d'argumentation proposé par Dung afin de doter les arguments d'une force. Le système d'argumentation proposé permet de considérer simultanément l'adhésion de plusieurs audiences via plusieurs classes d'acceptabilité et donc d'enrichir la sémantique du système d'argumentation.

Ainsi, le système d'argumentation (cf définition 2.3.1.I) est muni d'un ensemble de valeurs et d'un ensemble d'audiences. Plusieurs arguments peuvent prôner la même valeur.

Définition n° 2.3.3.I [Système d'argumentation à base de valeurs]

Un **système d'argumentation à base de valeurs** est un quintuplet

$VAS = \langle AR, attacks, V, promote, \Pi \rangle$ où :

- $AS = \langle AR, attacks \rangle$ est un système d'argumentation (cf définition 2.3.1.I) ;
- V un ensemble fini de valeurs $\{v_1, \dots, v_t\}$;
- $promote : AR \rightarrow V$ est une fonction qui met en relation les arguments et les valeurs. On dit qu'un argument P prône une valeur v ssi $promote(P) = v$;
- Π est un ensemble d'audiences $\{\alpha, \dots, \omega\}$;

Une audience correspond au concept d'audience formée proposé par Perelman (cf section 2.2.3). Les audiences se différencient les unes des autres de par les préférences qu'elles affectent aux valeurs. Il existe potentiellement autant d'audiences que d'ordres stricts sur l'ensemble des valeurs. On peut associer un système d'argumentation à chacune des audiences.

Définition n° 2.3.3.II [Système d'argumentation spécifique à une audience]

Un **système d'argumentation spécifique à une audience** α est un quintuplet $AAS = \langle AR, attacks, V, promote, \Pi, \succ_\alpha \rangle$ où $\alpha \in \Pi$ est l'audience en question et tq :

- $\langle AR, attacks, V, promote, \Pi \rangle$ est un système d'argumentation à base de valeurs (cf définition 2.3.3.I);
- \succ_α est une relation de préférences entre des valeurs c'est-à-dire une relation d'ordre strict sur V qui dépend de l'audience α .

$v_1 \succ_\alpha v_2$ se lit « v_1 est préféré à v_2 » selon l'audience α .

À partir de cette définition, on est amené à redéfinir le succès d'une attaque (cf définition 2.3.2.II) pour inclure l'audience concernée.

Définition n° 2.3.3.III [Défaire pour une audience]

Soit $AAS = \langle AR, attacks, V, promote, \Pi, \succ_\alpha \rangle$ un système d'argumentation spécifique à une audience α . Soient $P, Q \in AR$ deux arguments. P **défait** Q **pour une audience** α (noté $defeats_\alpha(P, Q)$) ssi :

$$attacks(P, Q) \wedge \neg(promote(Q) \succ_\alpha promote(P)).$$

Par abus de langage, on dit qu'un ensemble d'arguments S défait Q pour une audience α , ou bien Q est défait par S (noté $defeats_\alpha(S, Q)$) ssi un argument de S défait Q pour une audience α .

En d'autres termes, un argument Q est préféré à un argument P si la valeur prônée par l'argument Q est supérieure à la valeur prônée par l'argument A . On peut remarquer que le succès d'une attaque est garanti dans la mesure où les arguments prônent la même valeur. Si le système d'argumentation ne possède qu'une seule valeur, on retrouve la définition 2.3.1.I. Si chacun des arguments correspond à une valeur différente, on retrouve la définition 2.3.2.I.

La définition 2.3.2.III a ainsi été adaptée.

Définition n° 2.3.3.IV [Libre de conflit/Acceptabilité/Admissibilité]

Soit $AAS = \langle AR, attacks, V, promote, \Pi, \succ_\alpha \rangle$ un système d'argumentation spécifique à une audience α . Soient $P, Q \in AR$ deux arguments et $S \subseteq AR$ un ensemble d'arguments.

1. S est **libre de conflit** selon l'audience α ssi $\forall P, Q \in S \text{ defeats}_\alpha(P, Q)$;
2. P est **acceptable** selon l'audience α vis à vis de S (noté $acc_{VAS_\alpha}^S(A)$) ssi $\forall Q \in AR \text{ defeats}_\alpha(Q, P) \Rightarrow \text{defeats}_\alpha(S, Q)$;
3. S libre de conflit est **admissible selon l'audience α** (noté $adm_{VAS_\alpha}(S)$) ssi $\forall P \in S \text{ acc}_{VAS_\alpha}^S(P)$.

Comme dans la définition 2.3.1.V, une extension préférée est un ensemble admissible selon l'audience qui est maximal au sens de l'inclusion. On note $\text{pref}_{AAS_\alpha}(S)$ le prédicat correspondant.

On peut représenter un système d'argumentation spécifique à une audience au moyen d'un graphe orienté. Les sommets sont des arguments qui sont reliés par des arcs si l'argument représenté par le nœud source de l'arc défait selon cette audience l'argument représenté par le nœud destination de l'arc. La relation de préférences entre les valeurs étant une relation d'ordre strict, elle est asymétrique et transitive (cf annexe C). Un tel graphe ne possède pas de circuit. D'après le théorème 2.3.1.I, l'extension préférée correspond à l'extension raisonnable. Elle est unique et non-vide. De plus, le problème de décision associé est polynomial.

On peut considérer simultanément plusieurs audiences et donc enrichir la sémantique du système d'argumentation. On distingue trois nouvelles catégories d'acceptabilité, l'acceptabilité accordée par toutes les audiences (dite acceptabilité objective), l'acceptabilité accordée par quelques audiences (dite acceptabilité subjective) et l'acceptabilité accordée par aucune des audiences (dite acceptabilité indéfendable).

Définition n° 2.3.3.V [Acceptabilité objective/subjective/indéfendable]

Soit $VAS = \langle AR, attacks, V, promote, \Pi \rangle$ Un système d'argumentation à base de valeurs. Soit $P \in AR$ un argument et $S \subseteq AR$ un ensemble d'arguments :

- P est **acceptable objectivement** vis à vis de S ssi $\forall \alpha \in \Pi \text{ acc}_{VAS_\alpha}^S(P)$;
- P est **acceptable subjectivement** vis à vis de S ssi $\exists \alpha \in \Pi \text{ acc}_{VAS_\alpha}^S(P)$;
- P est **indéfendable** vis à vis de S ssi $\forall \alpha \in \Pi \neg \text{acc}_{VAS_\alpha}^S(P)$.

Les problèmes de décision liés à l'acceptabilité subjective et l'acceptabilité objective sont respectivement \mathcal{NP} -complet et $\text{co}\mathcal{NP}$ -complet .

Exemple n° 2.3.3.I [Système d'argumentation à base de valeurs] La figure 2.3.3.1 représente le système d'argumentation AAS_5 spécifique à une audience α . Si les

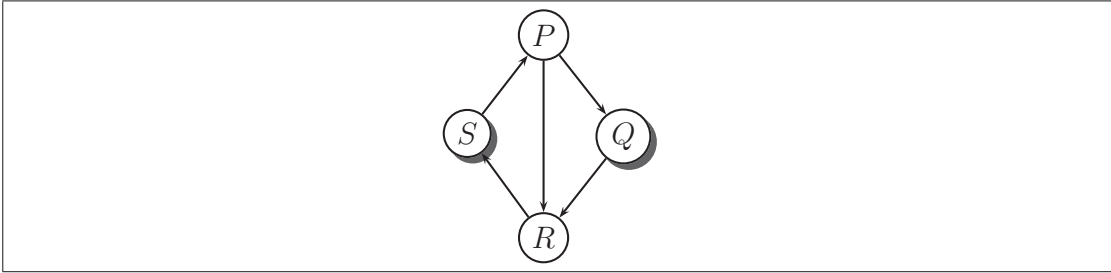


Fig. 2.3.3.I: Représentation du système d’argumentation AAS_5 spécifique à une audience α

noeuds ombrés prônent une valeur préférée par l’audience α , alors l’extension préférée $pref_{AAS_\alpha}(\{S, Q\})$, qui est également une extension raisonnable, est non-vide et unique. Dans le cas contraire, l’extension préférée $pref_{AAS_\alpha}(\{P, R\})$, qui est également une extension raisonnable, est non-vide et unique.

En résumé, un système d’argumentation abstrait est constitué d’un ensemble d’arguments muni d’une relation de contradiction (appelée attaque) et éventuellement d’une ou de plusieurs relations de priorité (appelée préférence). L’ensemble des travaux [21, 2, 9] présenté dans cette section ne fait pas état de la structure interne des arguments et se focalise sur la sémantique des systèmes argumentatifs. À l’inverse, un certain nombre de travaux ne considèrent pas les arguments comme des entités abstraites mais proposent une structure interne afin d’étudier les différentes relations qui les lient.

2.4 Logiques argumentatives

On appelle logique argumentative un système d’argumentation au travers duquel la structure des arguments (cf section 2.2.2) est définie à l’aide d’un langage logique sous-jacent. On distingue dans un argument la prémisse de la conclusion. Un argument est en faveur d’une conclusion si la conclusion est une conséquence logique de la prémisse.

La structure des arguments permet d’introduire différentes relations de contradiction. Dans un premier temps, nous allons présenter celles proposées par Michael Schroeder [51] (cf section 2.4.1). Les arguments peuvent être munis simultanément d’une structure et d’une force. Dans un second temps, nous nous tournerons vers la logique argumentative proposée par Leila Amgoud et Claudette Cayrol [1, 2] (cf section 2.4.2) qui s’appuie sur le système d’argumentation à base de préférences présenté précédemment. Finalement, nous présenterons la logique argumentative proposée par Antonis Kakas et Pavlos Moraitis [32, 42] (cf section 2.4.3) dans laquelle la force des arguments s’appuyant sur leur structure devient dynamique.

2.4.1 Relation de contradiction

Michael Schroeder [51] propose une logique argumentative. Un argument est défini comme un ensemble de clause de programme. À partir de cette structure, les arguments sont liés par un certains nombre de relations de contradictions. Ces relations sont comparées les unes aux autres.

La notion d'argument est formalisée à l'aide de programmes de logique étendue (*Extended Logic Programm*).

Définition n° 2.4.1.I [Programme de logique étendue]

Un **programme de logique étendue** est un ensemble de clauses de programmes du type :

$$r : L_0 \leftarrow L_1, \dots, L_m, \text{not}L_{m+1}, \dots, \text{not}L_{m+n}$$

avec L_i un littéral objectif ($\forall 0 \leq i \leq m+n$).

L_0 est appelé la **tête** de la clause de programme r . On le note $L_0 = \text{head}(r)$.

L'ensemble des autres littéraux constitue le **corps** de la clause. On le note : $\text{body}(r) = \{L_1, \dots, \text{not}L_{m+n}\}$.

Un programme de logique étendue comporte deux types de littéraux : des littéraux objectifs et des littéraux défauts. On appelle littéral **objectif** L , un atome A ou sa négation $\neg A$. On a alors explicitement $L = \neg\neg L$. Un littéral **défaut** est de la forme $\text{not}L^5$, avec L littéral objectif.

Dans un programme logique LP, étant donné un but G , l'arbre SLD de $\text{LP} \cup \{G\}$ peut avoir une branche infinie. Dans ce cas, il est impossible de déterminer si G est une conséquence logique du programme LP (noté $\text{not}G$). C'est la raison pour laquelle la négation par l'échec a été introduite.

Michael Schroeder [51] définit un argument comme un ensemble de clauses de programme.

⁵ $\text{not}L$ se lit : « Il n'y a aucune raison de croire en L ».

Définition n° 2.4.1.II [Arguments]

Soit LP un programme de logique étendue. Un **argument** P est une suite finie de clauses, $P = (r_1, \dots, r_i, \dots, r_n)$ définie telle que :

- $\forall i, 1 \leq i \leq n \ r_i \in LP$;
- $\forall i, 1 \leq i \leq n \ \forall L_j \in \text{body}(r_i) \ \exists k < i \ \text{tq} \ \text{head}(r_k) = L_j$;
- il n'y a pas deux clauses avec la même tête.

La tête d'une clause $\text{head}(r_i)$ est appelée **conclusion** et un littéral défaut dans l'une des clauses est appelé **hypothèse**.

L'ensemble des arguments de ce programme logique LP est noté AR_{LP} .

Un argument n'est plus une entité abstraite mais un ensemble de clauses de programme. La relation de contradiction de la définition 2.3.1.I peut alors être précisée.

Définition n° 2.4.1.III [Relations de contradiction]

Soit LP un programme de logique étendue. Une **relation de contradiction** sur LP est définie comme une relation $x_{LP} \subseteq AR_{LP} \times AR_{LP}$. Soient $P_1, P_2 \in AR_{LP}$ deux arguments.

1. P_1 **sape** P_2 (*undercuts*), noté $P_1 u_{LP} P_2$ ssi il existe un littéral objectif L défini tel que L est une conclusion de P_1 et $\text{not}L$ est une hypothèse de P_2 ;
2. P_1 **réfute** P_2 (*rebuts*), noté $P_1 r_{LP} P_2$ ssi il existe un littéral objectif L qui est une conclusion de P_1 et tel que $\neg L$ est une conclusion de P_2 ;
3. P_1 **attaque** P_2 (*attacks*), noté $P_1 a_{LP} P_2$ ssi P_1 contredit P_2 ou P_1 sape P_2 ;
4. P_1 **défait** P_2 (*defeats*), noté $P_1 d_{LP} P_2$ ssi P_1 sape P_2 ou P_1 contredit P_2 sans que P_2 ne sape P_1 ;
5. P_1 **attaque vigoureusement** P_2 (*strongly attacks*), noté $P_1 sa_{LP} P_2$ ssi P_1 attaque P_2 et P_2 ne sape pas P_1 ;
6. P_1 **sape vigoureusement** P_2 (*strongly undercuts*), noté $P_1 su_{LP} P_2$ ssi P_1 sape P_2 et P_2 ne sape pas P_1 .

Nous pouvons noter que la relation *rebut* est symétrique et que la relation *undercut* est antisymétrique. C'est à partir de ces deux relations que sont construites les autres relations.

Ces relations peuvent être généralisées quel que soit le programme de logique étendue en question : $x \in \{a, u, r, sa, su\}$. On peut alors ordonner ces relations binaires de la façon suivante : $x \subseteq y \Leftrightarrow \forall LP \ x_{LP} \subseteq y_{LP}$. Au même titre qu'une relation de contradiction peut être généralisée à n'importe quel programme logique, la relation de contradiction inverse ($x_{LP}^{-1} = \{(Q, P); (P, Q) \in x_{LP}\}$) est généralisable.

Par application des lois de la théorie des ensembles, on obtient le diagramme de Hasse de la figure 2.4.1, c'est à dire la représentation de l'ordre partiel sur les relations par sa réduction reflexo-transitive [51].

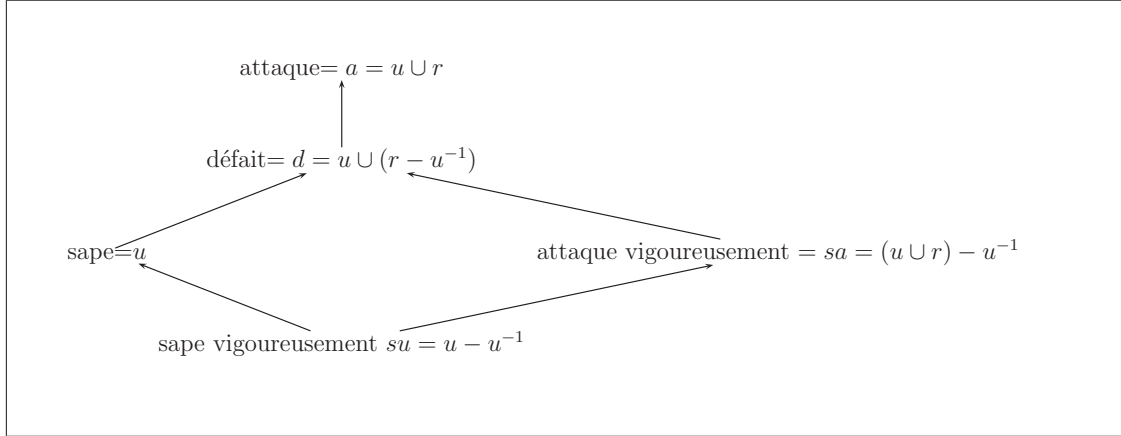


Fig. 2.4.1.I: Diagramme de Hasse des relations d'attaque

Déroulons l'exemple suivant pour illustrer ces relations de contradiction.

Exemple n° 2.4.1.I [Relation de contradiction] *Un adhérent d'une association doit, au cours d'une assemblée générale, se prononcer pour désigner le futur président de cette association. Sa connaissance est modélisable par un programme logique étendu LP qui contient les règles suivantes :*

- $r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$
- $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$
- $r_{12} : \text{prison}(x) \leftarrow \text{escroc}(x)$
- $r_{22} : \neg\text{president}(x) \leftarrow \text{prison}(x)$
- $r_3 : \text{escroc}(\text{jack})$
- $r_4 : \text{prison}(\text{jose})$
- $r_5 : \text{president}(\text{jack})$

Les règles r_{11} et r_{21} indiquent que cet adhérent doit choisir parmi deux candidats : jack et jose. Les règles r_{12} et r_{22} sont de sens commun : alors que la première signale que tous les escrocs vont en prison, la seconde énonce l'impossibilité d'être président lorsqu'on est en prison. Les trois règles r_3 , r_4 et r_5 expriment des opinions sur les compétences et les qualités des deux candidats.

On dispose des 3 arguments suivants :

- $P_{\text{jose}} = \{r_3, r_{12}, r_{22}, r_{21}\}$ qui est en faveur de la conclusion $\text{president}(\text{jose})$;

- $P_{jack} = \{r_4, r_{22}, r_{11}\}$ qui supporte la conclusion $president(jack)$;
- $P'_{jack} = \{r_5\}$ qui est en faveur de cette même conclusion.

On constate que $P_{jose}r_{LP}P_{jack}$ et $P_{jose}a_{LP}P_{jack}$. Par conséquence, $P_{jose}d_{LP}P_{jack}$ et $P_{jose}s_{LP}P_{jack}$. Les réciproques de ces relations sont vraies.

On constate également que $P_{jose}r_{LP}P'_{jack}$ et $P_{jose}a_{LP}P'_{jack}$. Par conséquence, $P_{jose}d_{LP}P'_{jack}$ et $P_{jose}s_{LP}P'_{jack}$. Là encore, les réciproques de ces relations sont vraies.

En résumé, la structure des arguments permet d'introduire différentes relations de contradiction. Le travail présenté dans la section suivante munit les arguments non seulement d'une structure mais également d'une force.

2.4.2 Relation de priorité statique

Dans [1, 2], Leila Amgoud et Claudette Cayrol proposent d'instancier le système d'argumentation à base de préférences présenté dans la section 2.3.2. La structure des arguments s'appuie sur une base de connaissances. Les arguments sont liés par des relations de contradictions et des relations de priorité (appelées préférences).

Les connaissances sont des jugements factuels. Ce sont des faits et des règles rassemblés dans une base de connaissances, notée Σ . Cette base contient des expressions bien formées du langage \mathcal{L} . Ce dernier est : soit un langage propositionnel, on parle alors de logique des propositions ; soit un langage du premier ordre, on parle alors de logique des prédicats. Nous utiliserons ici la syntaxe de la logique classique telle qu'elle a été définie dans l'annexe A. Quelle que soit la nature du langage envisagé, la déduction classique est notée \vdash et l'équivalence logique sera notée \equiv .

Les connaissances peuvent faire l'objet d'éventuelles inclinaisons, qu'elles soient liées à une vraisemblance, à une préférence ou à un principe de spécificité. À cet effet, la base est munie d'une relation d'ordre (partiel ou total) strict (cf annexe C). Nous notons \ll , cette relation sur Σ . L'objectif consiste à stratifier Σ en un ensemble fini de couches, c'est-à-dire de sous-ensembles disjoints notés $\Sigma_1, \Sigma_2, \dots, \Sigma_n$. La couche la plus haute Σ_1 contient les prédicats les plus préférables. À l'inverse, Σ_n est la couche la plus basse qui contient les formules les moins préférables. Par abus de notation, on écrit $\Sigma_n \ll \dots \ll \Sigma_1$. Cette stratification est telle que les formules de Σ_i sont équi-préférables et préférées à celles de Σ_j où $i \leq j$.

Cette notion de préférence peut être étendue à un ensemble de formules.

|| Définition n° 2.4.2.I [Niveau de préférence]

|| Soit (Σ, \ll) une base de connaissances munie d'une relation d'ordre strict. Soit H un sous-ensemble non vide de Σ . On appelle **niveau de préférence** de H , l'indice de la plus basse couche à laquelle appartient l'un des éléments de H .

L'exemple suivant vient illustrer ces définitions.

Exemple n° 2.4.2.I [Stratification d'une base de connaissances] *Comme dans l'exemple 2.4.1.1, un adhérent d'une association se prononce pour désigner le futur président de cette association. Il est muni de la base de connaissances Σ suivante :*

$$\Sigma_1 = \phi_1 = \textit{president}(\textit{jack}) \oplus \textit{president}(\textit{jose})$$

$$\Sigma_2 \supseteq \phi_{21} = \forall x \neg(\textit{president}(x) \wedge \textit{prison}(x))$$

$$\Sigma_2 \supseteq \phi_{22} = \forall x (\textit{escroc}(x) \rightarrow \textit{prison}(x))$$

$$\Sigma_3 = \phi_3 = \textit{escroc}(\textit{jack})$$

$$\Sigma_4 = \phi_4 = \textit{prison}(\textit{jose})$$

$$\Sigma_5 = \phi_5 = \textit{president}(\textit{jack})$$

Dans cette base de connaissances, une expression située au-dessus d'une autre expression est considérée comme plus prioritaire.

Étant données ces différentes définitions, nous pouvons définir un argument. Un argument est composé d'une formule appelée conclusion et d'un ensemble de formules, appelé support, à partir desquelles cette conclusion est inférée.

Définition n° 2.4.2.II [Argument]

Un **argument** est un couple $P = (H, h)$ où h est une formule de \mathcal{L} et H un sous-ensemble de Σ tel que :

1. $H \subseteq \Sigma$ est consistant ;
2. $H \vdash h$;
3. H est minimal, c'est-à-dire qu'il n'existe pas de sous-ensemble de H qui vérifie 1 et 2.

H est le support de P , noté $H = \textit{support}(P)$. h est la conclusion de P , notée $h = \textit{conclusion}(P)$.

L'ensemble des arguments construits sur cette base de connaissances est noté $\mathcal{A}(\Sigma)$.

Il est à noter qu'aucune hypothèse quant à la consistance de la base de connaissances n'a été émise. Ces éventuelles inconsistances peuvent être explicites lorsque deux formules de la base sont contradictoires ($p \in \Sigma$ et $\neg p \in \Sigma$) ou déduites lorsque les contradictions sont indirectes. L'exemple suivant illustre ce cas de figure.

Exemple n° 2.4.2.II [Argument] *On pourra noter que la base de connaissances présentée dans l'exemple 2.4.2.1 n'est pas consistante. D'une part, la formule suivante appartient à la base de connaissances :*

$$\textit{president}(\textit{jack}) \in \Sigma$$

D'autre part sa négation peut être déduite ainsi :

$$\begin{aligned} & \text{escroc}(\text{jack}), \\ & \forall x (\text{escroc}(x) \rightarrow \text{prison}(x)), \\ & \forall x \neg(\text{president}(x) \wedge \text{prison}(x)) \vdash \neg\text{president}(\text{jack}) \end{aligned}$$

On dispose alors de trois arguments dans $\mathcal{A}(\Sigma)$.

– un premier argument en faveur du candidat jack :

$$\begin{aligned} P_1 = & (\{\text{president}(\text{jack}) \oplus \text{president}(\text{jose}), \\ & \text{prison}(\text{jose}) \\ & \forall x \neg(\text{president}(x) \wedge \text{prison}(x))\}, \text{president}(\text{jack})) \end{aligned}$$

avec $\text{level}(\text{support}(P_1)) = 4$.

– un second argument en faveur du candidat jack :

$$P_2 = (\{\text{president}(\text{jack})\}, \text{president}(\text{jack}))$$

avec $\text{level}(\text{support}(P_2)) = 5$

– un troisième argument en faveur du candidat jose :

$$\begin{aligned} O_1 = & (\{\text{president}(\text{jack}) \oplus \text{president}(\text{jose}), \\ & \text{escroc}(\text{jack}), \\ & \forall x (\text{escroc}(x) \rightarrow \text{prison}(x)), \\ & \forall x \neg(\text{president}(x) \wedge \text{prison}(x))\}, \text{president}(\text{jose})) \end{aligned}$$

avec $\text{level}(\text{support}(O_1)) = 3$

Pour gérer de telles inconsistances, les arguments doivent alors être mis en relation. Ils sont liés les uns aux autres par des relations de contradictions et des relations de préférences.

La base de connaissances pouvant être inconsistante, les arguments sont susceptibles d'être en conflit. La première des relations présentées ici permet de capturer ces conflits.

|| Définition n° 2.4.2.III [sape]

Soient P_1 et P_2 deux arguments de $\mathcal{A}(\Sigma)$. P_1 **sape** P_2 ssi $\exists h \in \text{support}(P_2)$ tq $h \equiv \neg\text{conclusion}(P_1)$.

On note alors P_1 undercut P_2 .

La relation de « sape » est différente de celle définie par Michael Schroeder [51] (cf définition 2.4.1.III). Comme cela est formalisé dans la définition ci-dessus, un argument est sapé si l'un des éléments de son support est nié par un autre argument. Une seconde relation permet de départager des arguments en conflit les uns avec les autres.

Définition n° 2.4.2.IV [Préférence entre arguments]

P_1 est préféré à P_2 si et seulement si : $level(support(P_1)) > level(support(P_2))$.
On note alors $P_1 \text{ pref } P_2$.

Le niveau du support d'un argument constitue la force de l'argument.

C'est à partir de ces deux relations que le système d'argumentation à base de préférences est instancié (cf définition 2.3.2.I).

Définition n° 2.4.2.V [Système d'argumentation à base de préférences]

Un **système d'argumentation à base de préférences** est un triplet

$PAS = \langle \mathcal{A}(\Sigma), \text{undercut}, \text{pref} \rangle$ où :

- $\mathcal{A}(\Sigma)$ représente l'ensemble des arguments construits à partir de Σ , i.e. une base de connaissances qui contient des expressions bien formées du langage \mathcal{L} ;
- undercut est une relation de contradictions entre arguments (cf définition 2.4.2.III) ;
- pref est une relation de préférences, c'est-à-dire une relation d'ordre strict sur $\mathcal{A}(\Sigma)$ (cf définition 2.4.2.IV).

L'exemple suivant illustre cette instanciation.

Exemple n° 2.4.2.III [Relations entre arguments] Les arguments P_1, P_2 , et O_1 définis dans l'exemple 2.4.2.II sont liés les uns aux autres par des relations représentées dans la figure 2.4.2.I. À gauche, les relations de « sape » sont représentées à l'aide d'un graphe orienté. Au centre sont énumérées les relations de préférence. Le système d'argumentation à base de préférences est représenté par le graphe orienté à droite. Parmi

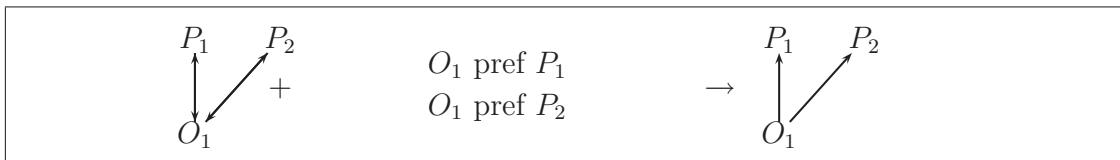


Fig. 2.4.2.I: Représentation du système d'argumentation à base de préférences

les arguments P_1, P_2 , et O_1 , seul O_1 est un argument acceptable vis à vis de $\mathcal{A}(\Sigma)$ (cf définition 2.3.2.III).

Afin de pouvoir évaluer la complexité algorithmique d'un tel système d'argumentation, nous devons définir au préalable un certain nombre de problèmes de décision associés. Le tableau 2.4.2.I synthétise la complexité algorithmique d'un certain nombre de problèmes de décision liés à un tel cadre d'argumentation dans le cas où \mathcal{L} est un langage logique du premier ordre. Le lecteur peu familier avec les classes de complexité peut se référer à l'annexe B. Ces résultats suffisent à montrer que ce cadre d'argumentation n'est pas tractable. Du moins dans le cas d'une logique du premier ordre.

En résumé, la logique argumentative présentée ici dote les arguments d'une structure interne mais également d'une force. Les priorités, qu'il s'agisse de vraisemblance, de préférence ou de spécificité sont statiques. Elles ne peuvent ni évoluer ni être remises en cause en cours de raisonnement. C'est l'objet des travaux présentés dans la section suivante.

2.4.3 Relation de priorité dynamique

Antonis Kakas et Pavlos Moraitis proposent dans [32, 42] une logique argumentative. Comme dans la section précédente, les arguments sont munis d'une structure interne et d'une force. Toutefois, cette force qui est justifiée à l'aide d'arguments peut évoluer au cours du raisonnement.

La notion d'argument est définie à l'aide de programme logique sans négation par l'échec (*Logic Program Without Negation as Failure*).

Définition n° 2.4.3.I [Programme logique sans négation par l'échec]

Un **programme logique sans négation par l'échec** est un ensemble de clauses de programmes du type :

$$r : L_0 \leftarrow L_1, \dots, L_n$$

avec L_i un littéral objectif ($\forall 0 \leq i \leq n$).

L_0 est appelé **tête** de la clause de programme r . On la note $L_0 = \text{head}(r)$.

L'ensemble des autres littéraux constitue le **corps** de la clause. On le note : $\text{body}(r) = \{L_1, \dots, L_n\}$.

Contrairement à la définition 2.4.1.I, un programme logique sans négation par l'échec comporte un seul type de littéral. Un littéral est un atome A ou la négation d'un atome $\neg A$.

Un programme logique sans négation par l'échec (noté LPWNaF) est muni d'une

Problème	Description	Complexité
$\text{pf}(h)$	Est-ce que $\exists H \subseteq \Sigma$ tq $H \vdash h$?	$\text{co}\mathcal{NP}$ -complet
$\text{pfc}(h)$	Est-ce que $\exists H \subseteq \Sigma$ tq $H \vdash h$ avec H consistante ?	$\Sigma_2\mathcal{P}$ -complet
$\text{pfc}_m(h)$	Est-ce que $\exists H \subseteq \Sigma$ tq $H \vdash h$ avec H consistante et minimale ?	$\Pi_2\mathcal{P}$ -complet
undercut (P)	Est-ce que l'argument P est sapé ?	$\Sigma_2\mathcal{P}$ -complet

Tab. 2.4.2.I: Complexité algorithmique des problèmes de décision liés à la relation de priorité statique

relation d'ordre strict (cf annexe C). Cette relation de priorité entre les règles peut être écrite sous la forme d'un atome.

Définition n° 2.4.3.II [Atome de priorité]

Soit $(LPWNaF, \ll)$ une relation d'ordre strict sur un programme logique sans négation par l'échec. L'atome *priority* est défini de la manière suivante :

$$\forall r_1, r_2 \in LPWNaF \text{ priority}(r_1, r_2) \Leftrightarrow r_2 \ll r_1$$

Une théorie argumentative est constituée d'un ensemble de règles à partir duquel on va sélectionner un sous-ensemble appelé argument. Un argument est en faveur d'une conclusion.

Définition n° 2.4.3.III [Théorie argumentative]

Soit (\mathcal{L}, \vdash) une logique monotone munie de la déduction classique. Une **théorie argumentative** est un couple $\mathcal{AT} = \langle \mathcal{T}, \mathcal{P} \rangle$ défini tel que :

- \mathcal{T} est une théorie c'est-à-dire un ensemble de clauses de programme (cf définition 2.4.3.I) appelées règles argumentatives ;
- \mathcal{P} est une théorie c'est-à-dire un ensemble de clauses de programme appelées également règles argumentatives dont les têtes de clauses peuvent être de la forme $L = \text{priority}(r_1, r_2)$ avec $r_1, r_2 \in \mathcal{T}$.

Un **argument** en faveur d'une conclusion L est un couple $(T, P) \in \mathcal{AT}$

tel que $T \cup P \vdash L$.

Afin d'éviter des problèmes d'auto-référence, aucune règle argumentative de \mathcal{T} ne contient dans son corps un littéral qui fait référence à l'atome de priorité. La relation de priorité est définie comme une partie de la théorie elle-même. Ainsi, on confère à la relation de priorité la même sémantique argumentative que le reste de la théorie.

On ne fait aucune hypothèse quant à la consistance de \mathcal{T} . Toutefois, on peut extraire des sous-ensembles de règles. Ces sous-ensembles constituent alors des arguments susceptibles d'être en conflit. La relation présentée ici permet de capturer ces conflits.

Définition n° 2.4.3.IV [Attaque]

Soit $\mathcal{AT} = \langle \mathcal{T}, \mathcal{P} \rangle$ une théorie argumentative. Soient $T, T' \subseteq \mathcal{T}$ et $P, P' \subseteq \mathcal{P}$.

(T, P) **attaque** (T', P') (noté $\text{attacks}((T, P), (T', P'))$) ssi

$\exists T_1 \subseteq T, T_2 \subseteq T'$ et $\exists P_1 \subseteq P, P_2 \subseteq P'$ tq :

1. $T_1 \cup P_1 \vdash L$ et $T_2 \cup P_2 \vdash \neg L$;
2. $\nexists T'_1 \subset T_1 \ T'_1 \vdash L$ et $\nexists T'_2 \subset T_2 \ T'_2 \vdash \neg L$;
3. $(\exists r \in T_1 \cup P_1, r' \in T_2 \cup P_2 \text{ tq } T \cup P \vdash \text{priority}(r', r)) \Rightarrow (\exists s \in T_1 \cup P_1, s' \in T_2 \cup P_2 \text{ tq } T' \cup P' \vdash \text{priority}(s, s'))$.

Comme cela est formalisé dans la définition ci-dessus, deux arguments se contredisent s'ils ont des conclusions contradictoires (condition 1), s'ils sont minimaux (condition 2) et s'ils ont la même force (condition 3). La relation d'attaque s'appuie sur la relation de conflit entre un littéral et sa négation explicite mais également sur la relation de priorité. Elle correspond à la relation de défaite (cf définition 2.3.2.II) dans un système d'argumentation à base de préférences. On peut noter que cette relation d'attaque est asymétrique.

C'est à partir de la relation d'attaque qu'est définie la notion d'admissibilité. Conformément à la définition 2.3.2.III, un argument est admissible dans la mesure où il est libre de conflit et capable de contre-attaquer les arguments qui l'attaquent.

Définition n° 2.4.3.V [Admissibilité]

Soit $\mathcal{AT} = \langle \mathcal{T}, \mathcal{P} \rangle$ une théorie argumentative. Soient $T \subseteq \mathcal{T}$, $P \subseteq \mathcal{P}$.

(T, P) **est admissible** ssi :

1. $T \cup P$ est consistant ;
2. $\forall T' \subseteq \mathcal{T}, P' \subseteq \mathcal{P} \text{ attacks}((T', P'), (T, P)) \Rightarrow \text{attacks}((T, P), (T', P'))$

Comme dans les travaux de Dung [21] (cf section 2.3.1) c'est à partir de cette notion d'admissibilité que sont définies les extensions ainsi que leur sémantique.

Définition n° 2.4.3.VI [Crédulité/Scepticisme]

Soit $\mathcal{AT} = \langle \mathcal{T}, \mathcal{P} \rangle$ un système d'argumentation. Soit L un littéral objectif.

L est une conséquence **crédule** (resp. **sceptique**) ssi L appartient à un (resp. à chaque) sous-ensemble admissible maximal de AS (au sens de l'inclusion ensembliste).

Adopter une nouvelle règle argumentative dans \mathcal{P} modifie les priorités. Puisque les priorités sont incorporées à la théorie argumentative, un agent peut adopter

une priorité au même titre qu'il adopte une règle argumentative. L'exemple 2.4.3.I permet d'illustrer ces définitions.

Exemple n° 2.4.3.I [Théorie argumentative] La figure 2.4.3.I représente la théorie argumentative $\langle \mathcal{T}, \mathcal{P} \rangle$. À gauche, sont énumérées les règles argumentatives de \mathcal{T} . À droite, sont énumérées les règles argumentatives de \mathcal{P} . Nous reprenons ici l'exemple

	$r_a : \text{priority}(r_{11}, r_{12})$
	$r_b : \text{priority}(r_{11}, r_{22})$
	$r_c : \text{priority}(r_{11}, r_{32})$
$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$	$r_d : \text{priority}(r_{21}, r_{12})$
$r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	$r_e : \text{priority}(r_{21}, r_{22})$
$r_{12} : \text{prison}(x) \leftarrow \text{escroc}(x)$	$r_f : \text{priority}(r_{21}, r_{32})$
$r_{22} : \neg\text{president}(x) \leftarrow \text{prison}(x)$	$r_g : \text{priority}(r_{12}, r_3)$
$r_{32} : \text{president}(x) \leftarrow \text{president_actuel}(x)$	$r_h : \text{priority}(r_{22}, r_3)$
$r_3 : \text{escroc}(\text{jack})$	$r_i : \text{priority}(r_{32}, r_3)$
$r_4 : \text{prison}(\text{jose})$	$r_j : \text{priority}(r_3, r_4)$
$r_5 : \text{president_actuel}(\text{jack})$	$r_k : \text{priority}(r_4, r_5)$
	$r_l : \text{menace}(\vee \neg\text{menace})$
	$r_m : \text{priority}(r_{32}, r_{22}) \leftarrow \neg\text{menace}$
	$r_n : \text{priority}(r_{32}, r_{12}) \leftarrow \text{menace}$

Fig. 2.4.3.I: Représentation d'une théorie argumentative

2.4.1.1 selon lequel un adhérent d'une association doit désigner le futur président de cette association. La modélisation est ici enrichie puisqu'on explicite la règle selon laquelle un président sortant doit être président (cf règle r_{32}). La priorité d'une telle règle dépend du contexte. Elle n'a pas la même priorité si l'association est menacée (règle r_n) ou non (règle r_m).

En cas d'absence de menace, les arguments admissibles en faveur du candidat jack sont :

- $(\{r_5, r_{32}\}, \emptyset)$;
- $(\{r_4, r_{22}, r_{11}\}, \{r_b\})$.

Il existe un unique argument admissible en faveur du candidat jose : $(\{r_3, r_{21}, r_{22}, r_{12}\}, \{r_e\})$. $\text{president}(\text{jack})$ et $\text{president}(\text{jose})$ sont toutes deux des conséquences crédules mais pas sceptiques.

En cas de menace, les arguments admissibles en faveur du candidat jack sont :

- $(\{r_4, r_{22}, r_{11}\}, \{r_b\})$;
- $(\{r_5, r_4, r_{22}, r_{32}\}, \{r_b, r_l\})$;
- $(\{r_5, r_{32}\}, \emptyset)$.

Il n'existe aucun argument admissible en faveur du candidat jose. president(jack) est une conséquence crédule et sceptique. president(jose) n'est ni une conséquence crédule ni une conséquence sceptique.

2.5 Synthèse

Comme nous l'avons souligné dans le chapitre 1, un agent doit être muni d'un modèle de raisonnement qui lui permet de gérer les incohérences entre ses propres croyances et les informations transmises par d'autres agents. Dans ce chapitre, nous avons montré que l'argumentation constitue un modèle adapté au processus cognitif d'un agent autonome et social pour gérer les interactions entre les arguments internes qui explicitent ses croyances et les arguments externes qui créditent des croyances contradictoires provenant d'autres agents.

Nous avons présenté ici deux approches philosophiques de l'argumentation. D'une part, Stephen Toulmin [57] fonde la rhétorique sur la notion d'évidence afin de donner une **structure aux arguments**. Un argument est un schéma qui permet d'aboutir à une conclusion. D'autre part, Chaïm Perelman et Lucie Olbrechts-Tyteca [15] fondent la rhétorique sur la notion d'adhésion afin de donner une **force aux arguments**. Convaincre quelqu'un, c'est vaincre par des arguments plus forts. Le consensus repose sur des valeurs et sur leurs hiérarchies. Ces travaux ont influencé la formalisation du raisonnement argumentatif.

Une logique argumentative s'appuie sur les cinq notions de base suivantes, qu'elles soient ou non explicitement formalisées :

1. **Un langage logique sous-jacent.** Les connaissances sont des jugements factuels qui peuvent être contradictoires. Ce sont des expressions bien formées d'un langage logique rassemblées dans une base de connaissances qui n'est pas nécessairement consistante.
2. **Le concept d'argument.** Un argument est : soit une entité abstraite, soit un schéma qui repose sur une base de connaissances. Dans le second cas de figure, on distingue dans un argument la prémisse de la conclusion. Un argument est en faveur d'une conclusion si la conclusion est une conséquence logique de la prémisse.
3. **La relation de contradictions entre arguments.** Puisque les connaissances peuvent être contradictoires, des arguments conflictuels peuvent co-exister. L'argumentation est une modalité naturelle pour raisonner à partir d'information contradictoire.
4. **Les relations de priorité entre les arguments.** Considérer deux arguments conflictuels, n'est pas suffisant pour pouvoir choisir parmi eux, il faut

prendre en considération toutes les informations pertinentes dont on dispose, notamment d'éventuelles inclinaisons, qu'elles soient liées à une vraisemblance, à une préférence, ou à un principe de spécificité. Ces relations dépendent de l'audience concernée. Qu'elles soient statiques ou dynamiques, elles dotent les arguments d'une force.

5. **La notion d'acceptabilité des arguments.** Les liens entre arguments permettent d'aboutir à une catégorisation des arguments appelée classes d'acceptabilité.

Néanmoins, l'argumentation ne constitue pas un modèle satisfaisant pour étudier la dynamique des flux informationnels dans un processus d'argumentation. Plusieurs questions restent ouvertes. Comment se déroule une argumentation ? Qui prend la parole ? Un argumentateur peut-il se répéter ? A-t-il le droit de se contredire, ou doit-il être consistant ? Qui a la charge de la preuve ? C'est l'objet du chapitre suivant.

“The true basis of the logic of existence and universality lies in the human activities of seeking and finding.”

The game of language by Jaakko Hintikka

Chapitre 3

Dialectique

Sommaire

3.1	Introduction	59
3.2	Système dialectique	60
3.3	Catégorisation des dialogues	64
3.4	Jeu logique d'évaluation sémantique	65
3.5	Jeu argumentatif d'évaluation sémantique	72
3.6	Synthèse	79

3.1 Introduction

En élargissant la vision décrite dans le chapitre précédent, l'argumentation peut être envisagée comme un dialogue entre deux opposants qui soutiennent des thèses contradictoires. Dans ce contexte, l'argumentation est un processus constitué par un ensemble d'interventions.

Afin d'illustrer notre propos, considérons un dialogue entre deux adhérents d'une association qui doivent désigner le futur président de cette association :

- P_1 : Jack ne peut pas être président.
- O_2 : Pourquoi ?
- P_3 : Jack devrait être en prison.
- O_4 : Pourquoi Jack devrait être en prison ?
- P_5 : Parce que Jack est un escroc.

Cette argumentation est un jeu à travers lequel deux participants, un partisan (*Proponent*) et un détracteur (*Opponent*), s'affrontent. Ils échangent des coups. Au terme de ce dialogue, la candidature de Jack est invalidée. En d'autres termes, le détracteur

est déclaré vainqueur puisque le partisan n'a pas réussi à défendre sa thèse contre toutes les attaques. La validité de cette thèse, c'est-à-dire la victoire du partisan, dépend des règles du jeu. Ces règles déterminent le tour de parole, la charge de la preuve, la pertinence des interventions... Elles permettent notamment d'éviter d'éventuels arguments circulaires (*petition principii*) et spécifient quand le jeu est terminé.

Cette mise en situation de l'argumentation est appelée dialectique. Le travail fondateur de Charles Hamblin, présenté dans la section 3.2, introduit la notion de système dialectique. En prolongeant cette démarche, Walton et Krabbe proposent une catégorisation des dialogues (cf section 3.3). Cette approche a permis de construire des systèmes dialectiques particuliers. Afin d'en introduire la formalisation, nous présenterons dans la section 3.4 un premier système dialectique qui permet de réifier la sémantique de la logique du premier ordre. Nous pourrions alors en présenter un second (cf section 3.5) qui en réifiant la sémantique d'un système d'argumentation montre que le dialogue peut être considéré comme un processus argumentatif.

3.2 Système dialectique

Charles L. Hamblin¹ [26] introduit dans son ouvrage fondateur intitulé *Fallacies* la notion de système dialectique.

Un **système dialectique** est une famille de dialogues régulés, *i.e.* un dispositif à travers lequel un ensemble de participants communiquent en respectant certaines règles. Dans le cas le plus simple, un dialogue se déroule entre deux parties qui prennent la parole chacune à leur tour. Elles interrogent et/ou répondent à leur partenaire via des locutions en prenant en compte les locutions précédentes.

Hamblin distingue la dialectique descriptive et la dialectique formelle. Aucune de ces deux approches ne prédomine l'autre. Elles viennent se nourrir mutuellement.

La **dialectique descriptive** est l'étude des systèmes de communication stylisée dans un contexte spécifiquement identifiable d'échanges linguistiques. On observe alors les règles sous-jacentes aux discussions réelles : débat parlementaire, instruction judiciaire, audition, ... Considérons, par exemple, un débat entre deux candidats à la veille d'un second tour d'une élection présidentielle. Aucun des participants ne doit se contredire, *i.e.* leur discours doit être consistant. Ils doivent ne pas se répéter, parler chacun à leur tour, signaler leur accord ou leur désaccord avec la précédente remarque de leur interlocuteur, avoir un temps de parole équivalent ...

¹Ce professeur de l'Université de Sydney est plus connu pour sa paternité du langage GEORGE, un des premiers langages à utiliser la notation polonaise inversée.

À l'inverse, la **dialectique formelle** vise l'élaboration d'un système simple² de règles précises qui ne sont pas nécessairement réalistes. Par exemple, Hamblin [26] imagine un système dialectique pour lequel chacun des participants doit donner son opinion sur l'affirmation précédente comme si c'était également une question. En d'autres termes, ce système dialectique est un système de question/réponse à travers lequel Alice interroge Bob qui doit fournir une réponse syntaxiquement correcte.

Le système dialectique proposé par Hamblin utilise deux notions essentielles : la notion de convention et la notion de tableau d'engagements.

Le dialogue doit se conformer à une **convention**, *i.e.* un ensemble de règles dialectiques. Les règles dialectiques sont contextuelles : elles dépendent de l'histoire passée du dialogue, le plus souvent seul le dernier coup est considéré. De plus, elles sont déontiques : elles spécifient les réponses qui sont autorisées et celles qui sont interdites. Un système est régulier (*rule-consistent*) si un coup n'est pas à la fois interdit et autorisé.

|| **Définition n° 3.2.0.VII [Règle dialectique]**

|| Soient C un contexte dialogique, S un ensemble de locutions et P un ensemble de participants. Une **règle dialectique** est de la forme :

|| « Si C est le cas alors l'ensemble des phrases S sont autorisées/interdites pour les participants P . »

|| S est l'objet de la règle et P la cible de la règle.

Les règles dialectiques peuvent être individuelles : la cible est un participant, ou elles peuvent être collectives : la cible est un ensemble de participants. Elles font référence à l'action locutoire (S) de ceux qui jouent un rôle dans le dialogue. Ces locutions peuvent être discriminatoires (elles s'adressent à un interlocuteur particulier) ou non (elles s'adressent à l'ensemble des participants). Dans le cas d'un dialogue entre deux participants, cette question est évincée.

Par exemple, le système dialectique proposé par Hamblin utilise la règle dialectique suivante : « Si la contribution d'Alice est de la forme p , alors Bob doit répondre p ou $\neg p$. » Cette règle est individuelle : elle est adressée à un interlocuteur en particulier.

Une règle dialectique peut contraindre un participant à ne pas se contredire. Il est alors dans l'obligation d'avoir un propos consistant. Formuler une telle règle dialectique n'est pas trivial. C'est la raison pour laquelle la notion de tableau d'engagements a été introduite.

²par opposition à complexe

On nomme **tableaux d'engagements** (*commitment stores*) les structures de données qui répertorient les engagements pris par les participants au cours du dialogue. Les tableaux d'engagements sont mis à jour en fonction des locutions émises afin d'enregistrer méticuleusement toutes les déclarations. Un joueur est cohérent dans son propos si son tableau d'engagements est consistant. Un joueur peut être cohérent dans son propos sans pour autant disposer de connaissances consistantes. Son tableau d'engagements ne correspond pas à ses croyances réelles qui sont privées, mais c'est un masque de croyances (*persona of beliefs*) qui est public. En pratique, nous ignorons bien souvent ce que sont réellement les croyances privées d'un argumentateur. Un système est sémantiquement cohérent lorsqu'un joueur n'est pas obligé d'énoncer une contradiction.

Par exemple, le système dialectique proposé par Hamblin oblige Bob à signaler son accord ou son désaccord vis à vis des remarques avancées précédemment par Alice. Bob construit un tableau d'engagements qui représente ses prises de position dans le dialogue.

Ainsi, Hamblin [26] distingue deux langages. Le contenu des locutions est conforme à un langage d'objets. Les règles dialectiques sont exprimées dans un langage de dialogues. Comme l'illustre l'exemple 3.2.0.II, un système dialectique permet de modéliser une interaction argumentative entre deux opposants qui soutiennent des thèses contradictoires.

Exemple n° 3.2.0.II [Dialogue] *Reprenons notre exemple de dialogue :*

- P_1 : Jack ne peut pas être président.
- O_2 : Pourquoi ?
- P_3 : Jack devrait être en prison.
- O_4 : Pourquoi Jack devrait être en prison ?
- P_5 : Parce que Jack est un escroc.

Dans le tableau 3.2.0.I, une locution émise par un des participants modifie non seulement son propre tableau d'engagements mais également le tableau d'engagements de son interlocuteur. Cela peut sembler contre-intuitif mais l'idée sous-jacente consiste à signaler que par défaut un participant est engagé sur ce que l'autre lui dit. C'est à l'interlocuteur d'exprimer son désaccord, s'il y a effectivement désaccord.

Un grand nombre de systèmes dialectiques ont été proposés dans la littérature : le système DC [31], le système PPD [67], le système RDP [67], ... Le lecteur intéressé peut se reporter à la thèse de Nicolas Maudet [39]. La démarche initiée par Hamblin permet de catégoriser les dialogues. C'est l'objet de la section suivante.

Tour	Joueur	Coup	CS_P	CS_O
1	P	assert(\neg president(jack))	\neg president(jack)	\neg president(jack)
2	O	challenge(\neg president(jack))	\neg president(jack)	? \neg president(jack)
3	P	assert(prison(jack))	\neg president(jack), prison(jack) prison(jack) \rightarrow \neg president(jack))	? \neg president(jack), prison(jack) prison(jack) \rightarrow \neg president(jack)
4	O	challenge(prison(jack))	\neg president(jack), prison(jack) prison(jack) \rightarrow \neg president(jack))	? \neg president(jack), ?prison(jack) prison(jack) \rightarrow \neg president(jack)
5	P	assert(escroc(jack))	\neg president(jack), prison(jack) prison(jack) \rightarrow \neg president(jack)) escroc(jack) \rightarrow prison(jack))	? \neg president(jack), ?prison(jack) prison(jack) \rightarrow \neg president(jack) escroc(jack) \rightarrow prison(jack)

Tab. 3.2.0.I: Mise à jour des tableaux d'engagements au cours du dialogue

3.3 Catégorisation des dialogues

Douglas N. Walton et Erik C. W. Krabbe ³ proposent dans [67] de catégoriser les dialogues. Leur typologie identifie 6 catégories primaires de dialogues.

D'après [67], un dialogue a pour but de faire évoluer une **situation initiale** pour atteindre les buts des participants. Ces buts peuvent être soit partagés par les participants, ils sont appelés alors **buts du dialogue**, soit être des objectifs qui leur sont propres. La situation initiale et les buts poursuivis permettent de distinguer 6 types de dialogues : la persuasion, la négociation, l'enquête, la délibération, la demande d'information et l'éristique.

Le tableau 3.3.0.II représente la grille d'analyse des dialogues proposée par Walton et Krabbe. Les six catégories de dialogues proposées se distinguent de par leur situation initiale et leur but principal.

Situation initiale → But principal ↓	Conflit	Problème ouvert	Ignorance d'un participant
Accord stable i.e. Résolution	persuasion	enquête	demande d'information
Règlement pratique i.e. Décision (ou non) d'agir	négociation	délibération	∅
Arrangement provisoire	éristique	∅	∅

Tab. 3.3.0.II: Aperçu systématique des types de dialogues

Dans un dialogue de **persuasion**, la situation initiale est un désaccord, c'est-à-dire un conflit d'opinion. Le but consiste à résoudre verbalement le conflit. Cela signifie que le processus dialectique est couronné de succès dans la mesure où l'un au moins des participants a changé d'opinion au cours du dialogue. Chacun essaie de persuader son interlocuteur d'adopter son point de vue. Les participants essaient d'établir ou de réfuter un argument pour conforter leur propre opinion. C'est un échange de propositions déclaratives.

Dans un dialogue de **négociation**, la situation initiale est un conflit d'intérêt mêlé à un besoin de coopération. Le but consiste à réaliser un échange ou conclure une affaire, c'est-à-dire trouver un compromis attractif pour les participants, chacun d'eux souhaitant maximiser son gain. C'est un échange de concessions.

Une **enquête** tâche d'établir (ou de prouver) la véracité de propositions dans le but de répondre à une question ouverte afin qu'un accord stable et général émerge. L'enquête est plus proche de la persuasion que de la négociation de par la nature

³Ils sont professeurs de Philosophie respectivement au Canada et au Pays-Bas.

déclarative des propositions. Toutefois une enquête se distingue d'une persuasion dans le sens où on ne débute pas par un conflit mais par un problème ouvert. L'ignorance initiale des participants est totale. Chacun d'eux souhaite étendre la somme de leurs connaissances.

Une **délibération**, comme une enquête, débute sur un problème ouvert et non pas sur un conflit. Le problème est, comme dans une négociation, pratique. Il consiste en une décision qui devra (ou non) être mise en œuvre. La discussion porte sur la finalité et les moyens d'une action dans le futur. Chacun des participants tente d'influencer le résultat.

Une **demande d'information** intervient lorsqu'un des participants souhaite disposer d'une connaissance qu'il suppose être détenue par son interlocuteur. La situation initiale est donc asymétrique. Ce n'est pas le cas d'une enquête où les participants sont tous ignorants. Le but consiste à diffuser cette connaissance. L'un des participants révèle son opinion.

Une **éristique** est une joute verbale qui consiste à humilier son interlocuteur autant que possible. La situation initiale est un conflit ou un antagonisme. Les participants ont été blessés et sont rancuniers. Un tel dialogue consiste en un échange de sentiments, de rancunes qui sont ouvertement reconnus et exprimés. L'objectif consiste à obtenir un arrangement provisoire dans une situation d'antagonisme où chacun des participants exprime le souhait de se comporter de manière plus sensible et faire preuve de plus de considération dans le futur.

Dans le monde réel, la nature des dialogues peut être mixte. Un dialogue peut donner lieu à différents sous-dialogues qui diffèrent de par leur nature. Le tableau 3.3 précise pour chacune de ces catégories de dialogues, la situation initiale et les buts poursuivis par les participants qu'ils soient partagés ou non. Quel que soit le type de dialogue envisagé, il consiste à révéler et à développer les positions des participants.

3.4 Jeu logique d'évaluation sémantique

La véracité d'une formule peut être établie au travers d'un système dialectique formel. Le jeu ainsi formalisé consiste en un dialogue de persuasion.

Au cours du vingtième siècle, des logiciens comme Paul Lorenzen⁴ [38] ou Jaakko Hintikka⁵ [28] pour ne citer qu'eux, entendent substituer à l'interprétation formaliste de la logique (cf annexe A.3) une sémantique nouvelle fondée sur le jeu logique d'évaluation sémantique. Ils s'efforcent d'assurer par là, à la logique elle-même, un

⁴Logicien et philosophe allemand (1915-1994)

⁵Logicien et philosophe finlandais (1929-)

Type du dialogue	Situation initiale	Buts partagés	But des participants
Demande d'information	ignorance d'un participant	diffusion d'information	donner/acquérir de l'information
Persuasion	conflit	accord stable	persuader l'interlocuteur
Enquête	ignorance des participants	accord stable	supporter/réfuter
Délibération	ignorance des participants	règlement pratique	évaluer finalité/moyens
Négociation	conflit d'intérêt besoin de coopération	règlement pratique	tirer le plus grand bénéfice
Éristique	conflit	arrangement provisoire	vaincre son adversaire

Tab. 3.3.0.III: Synthèse de la catégorisation des dialogues

fondement dialogique inspiré de nos activités argumentatives élémentaires. Au lieu d'utiliser des axiomes, des règles d'inférence, ou des tables de vérité, la conséquence logique est définie en terme de règles dialectiques.

Un jeu logique d'évaluation sémantique d'une formule ϕ dans une interprétation se joue entre deux participants communément appelés Éloïse et Abelard. Éloïse, appelée également vérificateur (noté \exists), affirme que l'interprétation est un modèle de ϕ . Abelard, appelé également falsificateur (noté \forall), affirme le contraire. Le jeu consiste en un dialogue de persuasion. Éloïse tente de convaincre Abelard de la véracité de ϕ . Abelard souhaite convaincre Éloïse du contraire. Le but du dialogue consiste à résoudre verbalement ce conflit.

La définition suivante décrit plus formellement un tel système dialectique :

Définition n° 3.4.0.VIII [Système dialectique]

Soient ϕ_0 une formule du premier ordre, M une interprétation et g_0 une fonction d'assignation ($g_0 : \mathcal{V} \rightarrow D$). Un **système dialectique** est un quintuplet $G_A(M, \phi_0, g_0) = \langle N, H, Z, T, (u_p)_{p \in N} \rangle$ défini tq :

- A est un ensemble d'actions : $A = \{(\psi, g)\}; \psi$ est une sous-formule de ϕ_0 et $g : \text{Free}(\psi) \rightarrow D$ une fonction d'assignation } ;
- $N = \{\forall, \exists\}$ est un ensemble de joueurs constitué d'Éloïse (notée \exists) et d'Abelard (noté \forall);
- H est l'ensemble des historiques, i.e. des séquences finies d'action et T une fonction de tour de parole. Ils sont définis récursivement sur la structure de ϕ_0 tq :

1. $h = (\phi_0, g_0) \in H$;

$$\forall n > 0 \ h = (h_0, \dots, h_{n-1}) \in H$$

- 2 si $h_{n-1} = (\psi \vee \theta, g)$ alors $\forall \chi \in \{\psi, \theta\}$, $(h_0, \dots, h_{n-1}, (\chi, g)) \in H$ et c'est au tour d'Éloïse de jouer : $T(h) = \exists$;

- 3 si $h_{n-1} = (\psi \wedge \theta, g)$ alors $\forall \chi \in \{\psi, \theta\}$, $(h_0, \dots, h_{n-1}, (\chi, g)) \in H$ et c'est au tour d'Abelard de jouer : $T(h) = \forall$;

- 4 si $h_{n-1} = (\neg\psi, g)$ alors $(h_0, \dots, h_{n-1}, (\psi, g)) \in H$ et $\exists \leftrightarrow \forall$;

- 5 si $h_{n-1} = (\exists x\psi, g)$ alors $\forall a \in D$, $(h_0, \dots, h_{n-1}, (\psi, g \cup \{(x, a)\})) \in H$ et c'est au tour d'Éloïse de jouer : $T(h) = \exists$;

- 6 si $h_{n-1} = (\forall x\psi, g)$ alors $\forall a \in D$, $(h_0, \dots, h_{n-1}, (\psi, g \cup \{(x, a)\})) \in H$ et c'est au tour d'Abelard de jouer : $T(h) = \forall$.

- L'ensemble Z des dialogues, i.e. des historiques de taille maximale.
- les gains $u_{\exists}, u_{\forall} : Z \rightarrow \{-1, 1\}$ sont des fonctions définies tq :

$$- u_{\exists}(h) = \begin{cases} 1, & \text{si } \langle M, g_{n-1} \rangle \models \phi_{n-1} \\ -1 & \text{sinon} \end{cases}$$

$$- u_{\forall}(h) = -u_{\exists}(h)$$

Le tour de parole et les règles dialectiques s'appuient sur la structure de ϕ .

1. Si ϕ est un atome, alors la satisfiabilité de ϕ dans l'interprétation détermine le résultat du jeu. Plus précisément, si l'interprétation est un modèle de ϕ alors Éloïse gagne sinon Abelard gagne.
2. Si ϕ est une disjonction, alors Éloïse choisit parmi cette disjonction une sous-formule et le jeu continue avec cette sous-formule.
3. Si ϕ est une conjonction, alors c'est à Abelard de choisir parmi cette conjonction une sous-formule et le jeu continue avec cette sous-formule.

4. Si ϕ est une négation, alors les joueurs inversent leur rôle et le jeu continue avec la négation de cette formule.
5. Si ϕ est préfixée par un quantificateur existentiel ($\phi : \exists x \psi(x)$), alors Éloïse sélectionne un objet d du domaine pour l'assigner à x et le jeu continue avec $\psi(d)$. C'est la raison pour laquelle Éloïse est notée \exists .
6. Si ϕ est préfixée par un quantificateur universel ($\phi : \forall x \psi(x)$), alors c'est à Abelard de sélectionner un objet d du domaine pour l'assigner à x et le jeu continue avec $\psi(d)$. C'est la raison pour laquelle Abelard est notée \forall .

D'après la définition 3.4.0.VIII, chaque dialogue est un historique terminal de la forme $h = (h_0, \dots, h_{n-1})$ avec $h_0 = (\phi_0, g_0)$ et $h_{n-1} = (\psi_{n-1}, g_{n-1})$ tel que ψ_{n-1} est une sous-formule atomique de ϕ_0 et $g_0 \subseteq g_1 \subseteq \dots \subseteq g_{n-1}$. Ainsi Z est l'ensemble des historiques qui ne peuvent pas être étendus.

On peut représenter un jeu à l'aide d'un arbre de décision. On parle alors de **représentation normale sous forme extensive**. Chaque nœud est une situation de jeu. Chacune des arêtes est associée à une action. À chaque nœud un des joueurs décide d'une action. Un joueur peut décider par avance de la meilleure action à entreprendre à tous les nœuds qu'il rencontre. Comme en théorie des jeux une stratégie, appelée également stratégie déterminée, désigne l'ensemble des choix d'actions décidés *a priori*. La définition suivante formalise cette notion dans le cas où le joueur envisagé est Éloïse.

Définition n° 3.4.0.IX [Stratégie (déterminée)]

Soit $G_A(M, \phi_0, g_0) = \langle N, H, Z, T, (u_p)_{p \in N} \rangle$ un système dialectique. $S_\exists \subseteq H$ est **une stratégie (déterminée)** pour Éloïse ssi il existe une fonction :

$$f_\exists : T^{-1}(\{\exists\}) \cap S_\exists \rightarrow A$$

tel que S_\exists est le plus petit ensemble qui satisfasse les conditions suivantes :

1. $(\phi_0, g_0) \in S_\exists$;
Si $\forall n > 0$ $h = (h_0, \dots, h_{n-1}) \in S_\exists$ alors
2. si $h_{n-1} = (\psi \wedge \theta, g)$ alors $\forall \chi \in \{\psi, \theta\}$, $(h_0, \dots, h_{n-1}, (\chi, g)) \in S_\exists$;
3. si $h_{n-1} = (\forall x \psi, g)$ alors $\forall a \in D$, $(h_0, \dots, h_{n-1}, (\psi, g \cup \{(x, a)\})) \in S_\exists$;
4. si $h_{n-1} \in \{(\exists x \psi, g), (\psi \vee \theta, g)\}$ alors $(h_0, \dots, h_{n-1}, f_\exists(h_0, \dots, h_{n-1})) \in S_\exists$.
Si $h_{n-1} = (\exists x \psi, g)$ alors la fonction f_\exists sélectionne un objet du domaine $a \in D$ pour étendre l'historique : $f_\exists(h_0, \dots, h_{n-1}) = (\psi, g \cup \{(x, a)\})$. Si $h_{n-1} = (\psi \vee \theta, g)$ alors la fonction f_\exists sélectionne l'une des sous-formules $\chi \in \{\phi, \psi\}$ pour étendre l'historique : $f_\exists(h_0, \dots, h_{n-1}) = (\chi, g)$.

Une stratégie (déterminée) pour Abelard peut être définie de manière similaire. Une stratégie gagnante pour Éloïse est une stratégie qui mène à un dialogue qu'elle gagne.

Définition n° 3.4.0.X [Stratégie gagnante]

Soit $G_A(M, \phi_0, g_0) = \langle N, H, Z, T, (u_p)_{p \in N} \rangle$ un système dialectique. Soit S_{\exists} une stratégie pour Éloïse. S_{\exists} est une **stratégie gagnante** pour Éloïse ssi :

$$\forall h \in S_{\exists} \cap Z \ u_{\exists}(h) = 1.$$

Une stratégie gagnante pour Abelard peut être définie de manière similaire. L'exemple suivant permet d'instancier ces définitions.

Exemple n° 3.4.0.III [Jeu logique d'évaluation sémantique]

Considérons une interprétation M dont le domaine est muni de deux objets s et t . Le jeu d'évaluation de la formule $\Phi : \forall x \exists y \ x \neq y$ se déroule de la manière suivante. Abelard commence par jouer. Il sélectionne un objet du modèle qu'il assigne à x . Éloïse lui répond. À son tour, elle assigne un objet du domaine à y . Quatre dialogues peuvent avoir lieu. Deux d'entre eux sont gagnés par Éloïse. Les deux autres sont gagnés par Abelard. Seule Éloïse a une stratégie gagnante dans ce jeu. Ce jeu est représenté sous forme normale extensive dans la figure 3.4.0.III.

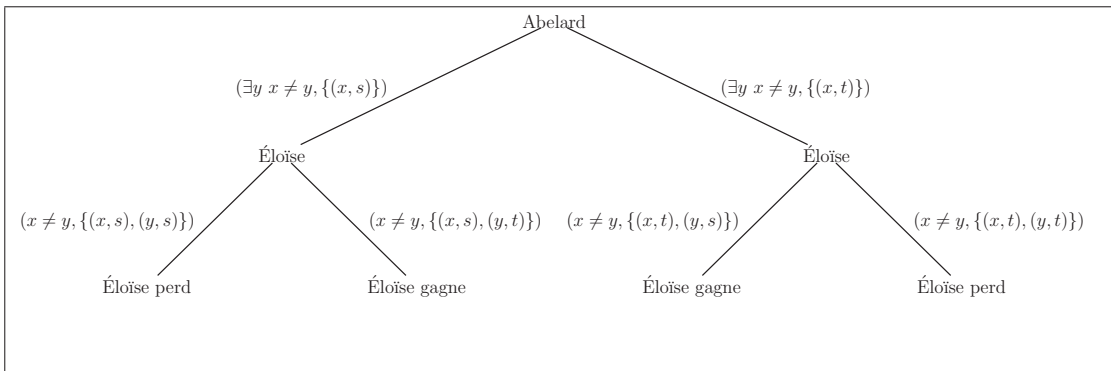


Fig. 3.4.0.I: Représentation sous forme normale extensive du jeu logique d'évaluation sémantique de la formule Φ dans l'interprétation M .

Un jeu logique d'évaluation sémantique est qualifié de déterminé, *i.e.* il existe une stratégie gagnante pour exactement un des joueurs. En effet, c'est un jeu à information parfaite, à somme nulle, pour deux joueurs, avec deux résultats de jeu et un nombre fini d'actions. D'après le théorème de Zermelo, ce type de jeu est déterminé [65].

Par conséquent, le jeu logique d'évaluation sémantique permet de réifier la sémantique de la logique du premier ordre. Une interprétation est un modèle pour une

formule si Éloïse a une stratégie gagnante dans le jeu correspondant. Dans le cas contraire, l'interprétation n'est pas un modèle de la formule.

Définition n° 3.4.0.XI [Satisfiabilité d'une formule]

Soient ϕ une formule du premier ordre, M une interprétation et g une fonction d'assignation ($g : \mathcal{V} \rightarrow D$). Soit $G_A(M, \phi, g) = \langle N, H, Z, T, (u_p)_{p \in N} \rangle$ le système dialectique associé.

ϕ est **satisfiable** dans $\langle M, g \rangle$ (noté $\langle M, g \rangle \models_{GTS}^+ \phi$) ssi il existe une stratégie gagnante pour Éloïse. On dit élément que $\langle M, g \rangle$ est un **modèle** de ϕ ;

ϕ n'est pas **satisfiable** dans $\langle M, g \rangle$ (noté $\langle M, g \rangle \not\models_{GTS}^- \phi$) ssi il existe une stratégie gagnante pour Abelard.

La proposition suivante met en adéquation la satisfiabilité au sens de Tarski ⁶ d'une formule du premier ordre et l'évaluation sémantique de cette même formule par le jeu correspondant.

Proposition n° 1 [Correction et complétude]

Soit ϕ une formule du premier ordre, M une interprétation et g une fonction d'assignation ($g : \mathcal{V} \rightarrow D$).

Correction : $\langle M, g \rangle \models_{GTS}^+ \phi \Rightarrow \langle M, g \rangle \models \phi$

$\langle M, g \rangle \models_{GTS}^- \phi \Rightarrow \langle M, g \rangle \not\models \phi$

Complétude : $\langle M, g \rangle \models \phi \Rightarrow \langle M, g \rangle \models_{GTS}^+ \phi$

$\langle M, g \rangle \not\models \phi \Rightarrow \langle M, g \rangle \models_{GTS}^- \phi$

Démonstration. Parce qu'un jeu logique d'évaluation sémantique est déterminé et en assumant l'axiome du choix, la démonstration se fait par induction sur la structure de ϕ . □

Nous avons vu ici que le jeu logique d'évaluation sémantique permet de réifier la sémantique de la logique du premier ordre. En d'autres termes, la satisfiabilité d'une formule dépend de l'existence d'une stratégie gagnante dans le jeu associé. Dans la section suivante, on introduit la notion de jeu argumentatif d'évaluation sémantique pour réifier la sémantique d'un système d'argumentation.

⁶*i.e.* au sens classique du terme comme défini dans l'annexe A.3.2

3.5 Jeu argumentatif d'évaluation sémantique

La théorie de la preuve dialectique proposée par Henry Prakken et Gerard Vreeswijk⁷ [66, 50] a pour but de réifier la sémantique d'un système d'argumentation (cf section 2.3.1). La sémantique est définie, non pas à l'aide de la notion d'extension, mais à l'aide d'un jeu, dans lequel s'affrontent un validateur et un falsificateur. Cette théorie assure à l'argumentation un fondement dialogique.

La théorie de la preuve dialectique identifie et formalise les principes qui gouvernent les échanges d'arguments et de contre-arguments sous la forme d'un jeu argumentatif d'évaluation sémantique. Les dialogues ainsi générés sont des persuasions particulières (cf section 3.3), appelés ici discussions critiques. Le but du validateur consiste à forcer le falsificateur à accepter sa thèse. Le but du falsificateur consiste à forcer le validateur à l'abandonner.

Un jeu argumentatif d'évaluation sémantique qui porte sur un argument P est joué par deux participants : un validateur qui soutient P et un falsificateur⁸ qui souhaite défaire P . Ils s'appuient tous deux sur le même système d'argumentation abstrait (cf définition 2.3.1.I). Ainsi, la structure des arguments n'est pas spécifiée (cf section 2.4). Les coups (*moves*) sont régulés par une fonction de tour de parole et des règles dialectiques. La définition suivante décrit plus formellement un tel système dialectique :

⁷Ces deux chercheurs sont actuellement membres du groupe *Intelligent Systems Group* du laboratoire d'informatique à l'université d'Utrecht.

⁸Dans ce manuscrit, nous utilisons le néologisme falsificateur pour désigner le joueur qui souhaite invalider l'hypothèse initiale. Un falsifieur est un maquilleur et un tricheur.

Définition n° 3.5.0.XII [Système dialectique]

Soit $AS = \langle AR, attacks \rangle$ un système d'argumentation. Soit $P \in AR$ un argument.

Un **système dialectique** qui porte sur P est un tuple

$G_{Moves}(P, AR, attacks) = \langle N, H, T, convention, Z, (u_p)_{p \in N} \rangle$ défini tq :

- $N = \{V, F\}$ est un ensemble de deux joueurs : un valideur V et un falsificateur F ;
- $Moves$ est un ensemble de coups bien formés. Un coup est soit un coup d'initialisation : $M_k = \langle player, Q \rangle$ soit un coup de réponse : $M_k = \langle player, Q, M_i \rangle$ tq le premier élément (noté $player(M_k)$) est un joueur, le second élément (noté $arg(M_k)$) est un argument et s'il y a lieu le troisième élément (noté $move(M_k)$) est un coup. Si $move(M_k) = M_i$, on dit que M_k répond à M_i ;
- H est l'ensemble des historiques, i.e. des séquences finies de coups $h_n = (M_1, \dots, M_n)$ défini tq $\forall 1 \leq k \leq n$:
 1. le joueur d'un coup est déterminé par une fonction de tour de parole : $player(M_k) = T(h_{k-1})$;
 2. les coups doivent être conformes à une convention : $M_k \in convention(h_{k-1})$.
- $T : H \rightarrow N$ est une fonction de tour de parole qui détermine étant donné un historique, le joueur du coup suivant. Si l'historique est de longueur nulle ou paire alors $T(h_n) = V$ sinon $T(h_n) = F$;
- $convention : H \rightarrow Moves$ est une fonction qui détermine étant donné un historique l'ensemble des coups autorisés.
- Z est l'ensemble des dialogues, i.e. les historiques de tailles maximales ;
- $u_V, u_F : H \rightarrow \{-1, 1\}$, i.e. les gains sont des fonctions éventuellement partielles qui déterminent le vainqueur d'un historique. Elles garantissent que :
 1. si un joueur gagne, l'autre joueur perd : $u_F(h_n) = -u_V(h_n)$;
 2. le joueur qui a le dernier mot gagne :

$$convention(h_n) = \emptyset \rightarrow \left((T(h_n) = V \rightarrow u_V(h_n) = -1) \wedge (T(h_n) = F \rightarrow u_V(h_n) = 1) \right)$$

D'après cette définition, chaque dialogue est de la forme $h_n = (M_1, \dots, M_n)$ avec $convention(h_n) = \emptyset$. Ainsi l'ensemble Z des dialogues est l'ensemble des historiques qui ne peuvent pas être étendus. La fonction $convention$ spécifie les coups autorisés (ou non).

Définition n° 3.5.0.XIII [Convention]

Soit $G_{\text{Moves}}(P, AR, attacks) = \langle N, H, T, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur P . La **convention** ($convention : H \rightarrow \text{Moves}$) est une fonction qui détermine étant donné un historique l'ensemble des coups autorisés. Elle garantit que :

1. le coup d'initialisation est joué par le valideur et porte sur P ($convention(\emptyset) = \langle V, P \rangle$);
2. un coup de réponse est autorisé dans la mesure où : d'une part il fait écho à un des coups précédents de son adversaire, d'autre part il comporte un argument qui attaque celui contenu dans le coup auquel il répond. $M_k \in convention(h_{k-1})$ (avec $k > 1$) si $\exists i, 1 \leq i < k$ tq :

$$move(M_k) = M_i \wedge player(M_k) \neq player(M_i) \wedge attacks(arg(M_k), arg(M_i))$$

De plus, deux coups qui répondent au même coup ne peuvent contenir le même argument : si $\exists j, i < j < k$ tq $move(M_j) = M_i$ alors $arg(M_k) \neq arg(M_j)$.

Le seul coup d'initialisation autorisé porte l'argument initial. Il est joué par le valideur. Les réponses sont autorisées si elles répondent à un des coups précédents de l'adversaire. L'argument avancé doit alors attaquer l'argument auquel il répond. De plus, deux coups qui répondent au même coup ne peuvent pas contenir le même argument.

En conséquence, les répétitions et les retours en arrière (*backtracks*) sont autorisés. La ligne persuasive d'un historique est la sous-séquence dans laquelle ces retours arrières ont été évincés.

Définition n° 3.5.0.XIV [Ligne persuasive]

Soient $G_{\text{Moves}}(P, AR, attacks) = \langle N, H, T, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique et $h_n = (M_1, \dots, M_n) \in H$ un historique. La **ligne persuasive** de h_n est la plus petite sous-séquence de h_n (noté $line(h_n)$) tq :

$$M_1, M_n \in line(h_n) \wedge \forall M_i \in line(h_n) M_i \in Legal(h_{i-1})$$

Le déroulement et le résultat d'un jeu logique d'évaluation sémantique (cf définition 3.4.0.VIII) sont définis par la structure de la formule sur laquelle porte le jeu. Dans un jeu argumentatif d'évaluation sémantique, on doit préciser la définition des gains et la définition de la convention.

Ainsi, une confrontation est un système dialectique à travers lequel deux joueurs répondent immédiatement au dernier coup (*TPI-dispute*, *Two-Party Immediate response dispute*) et où le falsificateur gagne lorsqu'il cite le valideur.

Définition n° 3.5.0.XV [Confrontation]

Soit $G_{\text{Moves}}(P, AR, attacks) = \langle N, H, T, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur P . Une **confrontation** est définie tq :

- la fonction *convention* est conforme à la définition 3.5.0.XIII. Étant donné un historique $h_n = (M_1, \dots, M_n)$ et M_k un coup dans l'historique (avec $1 \leq k \leq n$), la convention détermine les coups de réponse autorisés de la manière suivante :
 - les deux joueurs V et F sont autorisés à répéter les arguments de V :
 $M_{n+1} \in convention(h_n)$ si $arg(M_{n+1}) = arg(M_k)$ et $player(M_k) = V$
 - V n'est pas autorisé à répéter les arguments de F :
 $M_{n+1} \notin convention(h_n)$ si $arg(M_{n+1}) = arg(M_k)$,
 $player(M_k) = F$ et $player(M_{n+1}) = V$
 - F est autorisé à répéter ses propres arguments dans la mesure où ils ne sont pas dans la même ligne persuasive :
 $M_{n+1} \notin convention(h_n)$ si $arg(M_{n+1}) = arg(M_k)$, $M_k \in line(h_n)$
 $player(M_{n+1}) = F$ et $player(M_k) = F$
- les gains u_V et u_F sont conformes à la définition 3.5.0.XII. Ils sont définis tq si le falsificateur cite le valideur alors le falsificateur gagne :
 si $arg(M_{n+1}) = arg(M_k)$, $player(M_k) = V$ et $player(M_{n+1}) = F$
 alors $u_V(h_{n+1}) = -1$

Comme dans un jeu logique d'évaluation sémantique, une stratégie désigne un ensemble de choix d'action décidés *a priori*. La définition suivante formalise cette notion dans le cas où le joueur envisagé est le valideur.

Définition n° 3.5.0.XVI [Stratégie (déterminée)]

Soit $G_{\text{Moves}}(P, AR, attacks) = \langle N, H, T, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur P . $S_V \subseteq H$ est **une stratégie (déterminée)** pour le valideur ssi il existe une fonction :

$$f_V : T^{-1}(\{V\}) \cap S_V \rightarrow \text{Moves}$$

tel que S_V est le plus petit ensemble qui satisfasse les conditions suivantes :

1. si $T(M_1, \dots, M_{k-1}) = F$ alors $(M_1, \dots, M_k) \in S_V$;
2. si $T(M_1, \dots, M_{k-1}) = V$ alors $(M_1, \dots, M_{k-1}, f_V(M_1, \dots, M_{k-1})) \in S_V$, la fonction f_V sélectionnant le coup de réponse M_k pour étendre l'historique.

Une stratégie pour le falsificateur peut être définie de manière similaire. Une stratégie gagnante pour le valideur est une stratégie qui mène à un historique qu'il gagne. Ainsi, la définition 3.4.0.X a été modifiée de la manière suivante :

Définition n° 3.5.0.XVII [Stratégie gagnante]

Soit $G_{\text{Moves}}(P, AR, \text{attacks}) = \langle N, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur P . Soit S_V une stratégie pour le valideur. S_V est une **stratégie gagnante** pour le valideur ssi :

$$\forall h_n \in S_V \cap H \ u_V(h_n) = 1.$$

Une stratégie gagnante pour le falsificateur peut être définie de manière similaire. Pour les mêmes raisons que celles évoquées dans la section précédente, le jeu est déterminé. Par conséquent, la confrontation permet de réifier la sémantique crédule et la sémantique sceptique du système d'argumentation associé (cf définition 2.3.1.VI).

Définition n° 3.5.0.XVIII [Prouvabilité crédule/sceptique]

Soient $AS = \langle AR, \text{attacks} \rangle$ un système d'argumentation et $P \in AR$ un argument. Soit $G_{\text{Moves}}(P, AR, \text{attacks}) = \langle N, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ la confrontation associée.

On dit de P qu'il est **prouvable avec crédulité** dans AS (noté $\text{provable}_c^{\text{AS}}(P)$) ssi il existe une stratégie gagnante pour le valideur dans la confrontation associée. Un argument n'est pas prouvable avec crédulité dans AS (noté $\neg\text{provable}_c^{\text{AS}}(P)$) ssi il existe une stratégie gagnante pour le falsificateur dans la confrontation associée.

On dit de P qu'il est **prouvable avec scepticisme** dans AS (noté $\text{provable}_s^{\text{AS}}(P)$) ssi il est prouvable avec crédulité dans AS et que toutes ses attaques ne sont pas prouvables avec crédulité dans AS : $\forall Q \in AR$ tq $\text{attacks}(Q, P)$, $\neg\text{provable}_c^{\text{AS}}(Q)$.

Les deux propositions suivantes attestent de la correction et de la complétude de la confrontation vis à vis de la sémantique crédule d'une part et vis à vis de la sémantique sceptique d'autre part.

Proposition n° 2 [Prouvabilité avec crédulité]

Soit $AS = \langle AS, \text{attacks} \rangle$ un système d'argumentation et $P \in AR$ un argument.

$$\text{Correction : } \text{provable}_c^{\text{AS}}(P) \Rightarrow c(P, AS)$$

$$\text{Complétude : } c(P, AS) \Rightarrow \text{provable}_c^{\text{AS}}(P)$$

Prouvabilité avec crédulité. Les extensions préférées sont des ensembles admissibles maximaux pour l'inclusion ensembliste (cf définition 2.3.1.V). En conséquence, un argument appartient à au moins une extension préférée ssi il appartient à un ensemble admissible.

- *Correction.* Supposons que $\text{provable}_c^{\text{AS}}(P)$. Soit S l'ensemble des arguments avancés par V pour prouver P . D'une part S est libre de conflit. Dans le cas contraire, il existe $Q, R \in S$ tel que $\text{attacks}(Q, R)$. Ainsi F gagne en citant l'argument Q avancé par V qui attaque R . Ce n'est pas le cas. D'autre part, S est admissible. Dans le cas contraire, il existe $O \in \text{AR}$ tq $\text{attacks}(O, S)$ et $\neg \text{attacks}(S, O)$. Ainsi F gagne en utilisant cet argument. Ce n'est pas le cas.
- *Complétude.* Supposons maintenant que $P \in S$ avec S un ensemble admissible d'arguments. Ainsi V peut utiliser les arguments de S pour défendre P . Chacun des arguments de S peut être défendu contre toutes les attaques par un argument de S . F ne peut citer les arguments de S avancés par V puisque cet ensemble est libre de conflit.

□

Si le système d'argumentation est cohérent (cf définition 2.3.1.VIII), la prouvabilité avec scepticisme d'un argument dans un système d'argumentation peut être mise en adéquation avec la sémantique sceptique de ce même argument (cf définition 2.3.1.VI).

Proposition n° 3 [Prouvabilité avec scepticisme]

Soit $\text{AS} = \langle \text{AS}, \text{attacks} \rangle$ un système d'argumentation cohérent. Soit $P \in \text{AR}$ un argument.

$$\text{Correction} : \text{provable}_s^{\text{AS}}(P) \Rightarrow s(P, \text{AS})$$

$$\text{Complétude} : s(P, \text{AS}) \Rightarrow \text{provable}_s^{\text{AS}}(P)$$

Prouvabilité avec scepticisme. Supposons que le système d'argumentation est cohérent. Ainsi toutes ses extensions préférées sont également stables (cf définition 2.3.1.VIII).

- *Complétude.* Soit P un argument dans l'intersection des extensions préférées. Tous les arguments Q qui attaquent P sont attaqués par ces extensions. En conséquence, P est prouvable avec crédulité. De plus, Q ne peut appartenir à de telles extensions qui sont libres de conflits. Par conséquence, Q n'est pas prouvable avec crédulité.
- *Correction.* Supposons maintenant que P est prouvable avec crédulité et qu'aucun des arguments Q qui l'attaquent n'est prouvable avec crédulité. Ainsi, aucune de

ces attaques n'appartient à une extension préférée qui est par hypothèse également une extension stable. Par conséquent, P est défendu par l'ensemble de ces extensions qui contiennent toutes P .

□

L'exemple suivant permet d'instancier ces définitions.

Exemple n° 3.5.0.IV [Prouvabilité avec crédulité et scepticisme] *Le système d'argumentation AS_2 est représenté dans la figure 3.5.0.II. La confrontation associée à*

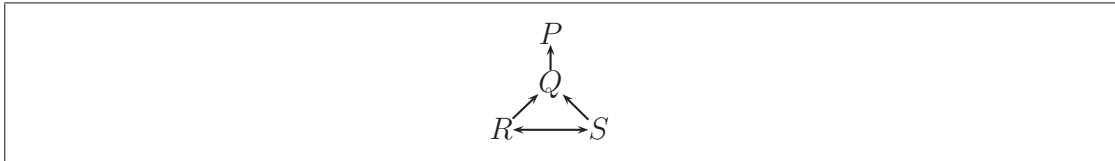


Fig. 3.5.0.II: Représentation sous forme de graphe orienté du système d'argumentation AS_2 .

AS_2 qui porte sur P (resp. Q) est représentée à gauche (resp. à droite) dans la figure 3.5.0.III. Concernant la confrontation qui porte sur P , les deux dialogues possibles sont gagnés par le valideur. En effet, le falsificateur n'est pas autorisé à se répéter lui-même. La confrontation qui porte sur Q peut donner lieu quant à elle à deux dialogues gagnés par le falsificateur. Ce dernier cite le valideur pour attaquer Q . AS_2 est cohérent. En

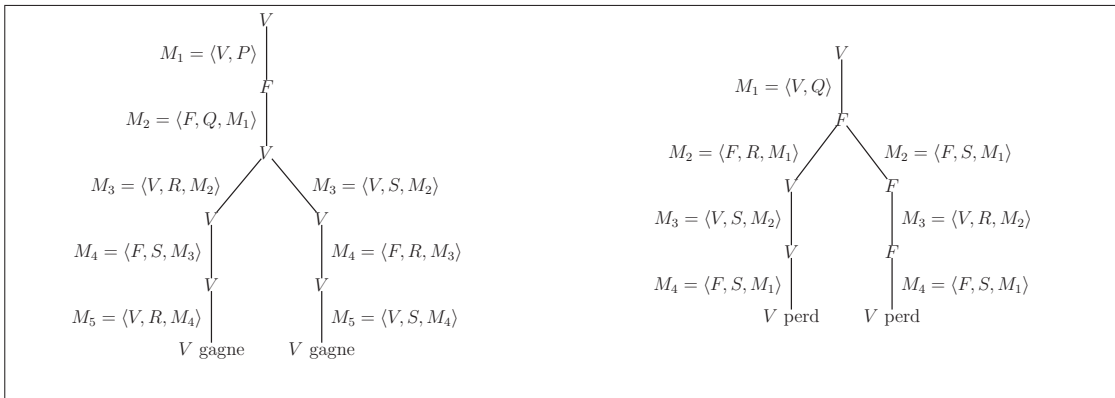


Fig. 3.5.0.III: Représentation sous forme normale extensive des confrontations sur P (à gauche) et Q (à droite) dans le système d'argumentation AS_2

effet, il possède deux extensions préférées : $pref_{AS_2}(\{P, R\})$ et $pref_{AS_2}(\{P, S\})$ qui sont également des extensions stables. Toutes deux contiennent P . Cet argument est donc accepté par un agent rationnel qu'il soit sceptique ou crédule. On vérifie sur la figure 3.5.0.III que P est prouvable avec scepticisme. En effet P est prouvable avec crédulité et Q n'est pas prouvable avec crédulité.

En résumé, la confrontation est un jeu argumentatif d'évaluation sémantique à travers lequel s'affrontent un validateur qui affirme qu'un argument est valide et un falsificateur qui affirme le contraire. Un tel jeu permet de réifier la sémantique du système d'argumentation associé.

3.6 Synthèse

Dans ce chapitre, nous avons envisagé l'argumentation comme un dialogue entre deux opposants qui soutiennent des thèses contradictoires. Cette mise en situation de l'argumentation est appelée dialectique.

Un **système dialectique** est un dispositif à travers lequel un ensemble de participants jouent. D'une part, un dialogue est régulé par une **convention**, i.e. un ensemble de règles dialectiques. Ces règles spécifient, étant donné un contexte dialogique, les locutions qui sont autorisées (ou non). D'autre part, les déclarations sont stockées au fur et à mesure dans des structures de données intitulées **tableaux d'engagements**.

Un dialogue est un jeu qui a pour but de faire évoluer une **situation initiale** pour atteindre les buts des participants. Ces buts peuvent être soit partagés par les participants, ils sont appelés alors **buts du dialogue**, soit des objectifs qui leur sont propres. La situation initiale et les buts poursuivis permettent de distinguer 6 types de dialogues : la persuasion, la négociation, l'enquête, la délibération, la demande d'information et l'éristique.

Les préoccupations des dialecticiens dépassent celles des logiciens. La dialectique est une étude plus générale que la logique, en ce sens que la logique peut être conçue comme un système dialectique. À cet effet, nous avons envisagé deux systèmes dialectiques particuliers.

Un jeu logique d'évaluation sémantique est un jeu persuasif particulier à travers lequel s'affrontent un validateur qui affirme qu'une formule est satisfiable et un falsificateur qui affirme le contraire. Ce jeu, qui réifie la sémantique de la logique du premier ordre, montre comment un système dialectique peut être formalisé.

Un jeu argumentatif d'évaluation sémantique est un jeu persuasif particulier à travers lequel s'affrontent un validateur qui affirme qu'un argument est valide et un falsificateur qui affirme le contraire. Ce jeu, qui permet de réifier la sémantique d'un système d'argumentation, conforte l'idée selon laquelle un dialogue peut être considéré comme un processus argumentatif.

Nous allons dans le chapitre suivant voir dans quelle mesure ces travaux ont influencé les modèles d'interaction entre agents.

“All coordination mechanisms can ultimately be reduced to (joint) commitments and their associated (social) conventions”.

Commitments and Conventions : The Foundation of Coordination in Multi-Agent Systems by Nick R. Jennings

Chapitre 4

Modèle de dialogue entre agents

Sommaire

4.1	Introduction	81
4.2	Dialogue et engagements	82
4.2.1	Dialogue pour la formation d'équipe	82
4.2.2	DIAGAL	85
4.3	Dialogue et arguments	88
4.3.1	Système multi-agents argumentatifs	88
4.3.2	Système dialectique multi-agents	91
4.4	Modèle de dialogue	108
4.4.1	Le modèle de dialogue	108
4.4.2	Les relations entre engagements et arguments	109
4.5	Synthèse	111

4.1 Introduction

Dans ce chapitre, nous allons montrer dans quelle mesure la dialectique, présentée dans le chapitre 3, a influencé les modèles d'interaction entre agents. La dialectique fut à l'origine d'un très grand nombre de travaux dans le domaine des SMA. Il est donc difficile de prétendre à l'exhaustivité. On peut toutefois citer parmi ceux qui ont été à l'origine de cette investigation, le travail de Chris Reed [52] ou celui de Nick Jennings, Carlos Sierra et Simon Parsons [45]. Contrairement à ceux-ci, d'autres travaux plus récents qui s'appuient sur ces derniers seront présentés ici.

Parce que la notion d'engagement est fondamentale dans la dialectique, nous allons voir dans la section 4.2 dans quelle mesure cette notion a été intégrée dans les modèles d'interaction directe entre agents. Selon la dialectique, le dialogue peut être

considéré comme un processus argumentatif. Nous allons donc présenter dans la section 4.3 des modèles de dialogue entre agents qui s'appuient davantage sur la notion d'argument que sur la notion d'engagement. Nous terminerons ce chapitre par la présentation d'un modèle de dialogue entre agents qui explicite le lien qui existe entre les engagements pris au cours des dialogues et les arguments échangés (cf section 4.4).

4.2 Dialogue et engagements

La notion d'engagement est fondamentale dans la dialectique. Cette notion permet justifier la pertinence de la dialectique pour les SMA (cf section 4.2.1). La section 4.2.2 présente un modèle de dialogue entre agents qui s'appuie sur la notion d'engagement.

4.2.1 Dialogue pour la formation d'équipe

Frank Dignum, Barbara Dunin-Keplicz et Rineke Verbrugge proposent dans [19, 20] un cadre générique et flexible pour la résolution coopérative de problèmes. Les agents participent à l'élaboration d'une équipe au travers de dialogues de différentes natures dans le but d'établir des engagements sociaux.

Dans la section 4.2.1.1, nous allons montrer dans quelle mesure la théorie du dialogue permet de proposer un processus de résolution flexible. Nous présenterons ensuite le formalisme utilisé pour modéliser les différentes stratégies d'engagement (cf section 4.2.1.2).

4.2.1.1 Cadre flexible de résolution collaborative de problème

Le protocole « Contract Net » [61] présenté dans la section 1.4.1.2 permet de gérer un processus pour la résolution coopérative de problèmes. Ce processus est simple et efficace. Toutefois il se révèle être trop rigide. Les caractéristiques de la tâche ne peuvent pas être discutées par les participants. Elles sont fixées par l'appel d'offre et ne peuvent pas être modifiées au cours du processus.

Le cadre générique et flexible pour la résolution coopérative de problèmes proposé dans [19, 20] subdivise le processus de résolution en quatre étapes :

1. **identification potentielle** (*potential recognition*), l'agent à l'initiative de la résolution cherche des partenaires pour collaborer au sein d'une équipe afin de résoudre ensemble le problème. L'initiateur récolte des précisions concernant les candidats potentiels à l'aide de dialogues de demande d'information ;

2. **formation d'équipe** (*team formation*), l'ensemble des agents expriment leur volonté de participer (ou non). L'initiateur tente de convaincre les candidats de participer à l'équipe à l'aide de dialogues de persuasion ;
3. **formation de plan** (*plan formation*), l'équipe d'agents subdivise le but en différentes tâches. Ces tâches sont partagées et allouées aux membres de l'équipe. Les participants se distribuent les tâches à effectuer à l'aide de dialogues de négociation ;
4. **exécution de plan** (*plan execution*), les membres de l'équipe exécutent les tâches qui leur ont été allouées. L'exécution est supervisée et peut être replanifiée.

Ce cadre pour la résolution coopérative de problèmes peut être qualifié de flexible dans la mesure où il ne s'appuie pas sur le seul protocole « Contract Net » mais sur des dialogues de différentes natures. Au cours de ces dialogues, les agents prennent des engagements (sociaux) en fonction de leur stratégie.

4.2.1.2 Formalisation des stratégies d'engagements

Le cadre proposé par [19, 20] ne permet pas de spécifier complètement le processus de résolution. Toutefois, il formalise à l'aide de la logique CTL la notion de stratégie d'engagement qui est essentielle à sa mise en œuvre.

La logique CTL (*Computation Temporal Logic*) est une logique temporelle arborescente. Les formules CTL s'appliquent à des systèmes de transitions étiquetées par des actions et qui énoncent des propriétés d'états. Les formules utilisant les opérateurs temporels peuvent être interprétées de la manière suivante :

- $\text{inevitable}(\psi)$: ψ est vrai sur tous les chemins à partir du point de référence ;
- $\text{optional}(\psi) \equiv \neg \text{inevitable}(\neg \psi)$: ψ est vrai sur l'un des chemins ;
- $\diamond \psi$: ψ est vrai quelque part sur un chemin ;
- $\psi U \phi$: ψ est vrai. Dès que cela n'est plus le cas, ϕ est vrai.

Afin de caractériser les candidats, le langage est muni de trois prédicats (*able*, *opp* et *willing*). Les formules utilisant ces prédicats peuvent être interprétées de la manière suivante :

- $\text{able}(b, \phi)$: l'agent b a la capacité individuelle d'atteindre le but ϕ ;
- $\text{opp}(b, \phi)$: l'agent b a l'opportunité d'atteindre ϕ ;
- $\text{willing}(b, \phi)$ l'agent b a la volonté d'atteindre ϕ .

Le prédicat $\text{COMM}(a, b, \psi)$ signale que l'agent a s'engage socialement, *i.e.* publiquement, vis à vis de l'agent b à remplir ψ .

Différents niveaux de persistance des intentions (cf section 1.3.3) peuvent être capturés en différenciant plusieurs **stratégies d'engagement** :

- la stratégie d'**engagement aveugle** (*blindly committed*), l'agent a conserve son engagement jusqu'à ce qu'il soit atteint :

$$\begin{aligned} \text{COMM}(a, b, \psi) \rightarrow \\ \text{inevitable}[\text{COMM}(a, b, \psi)U \\ \{\text{BEL}(a, \psi)\}] \end{aligned} \quad (4.2.1.I)$$

- la stratégie d'**engagement simple** (*single-minded*), l'agent a abandonne son engagement lorsqu'il pense que cet engagement n'est plus réalisable :

$$\begin{aligned} \text{COMM}(a, b, \psi) \rightarrow \\ \text{inevitable}[\text{COMM}(a, b, \psi)U \\ \{\text{BEL}(a, \psi) \vee \\ (\neg \text{BEL}(a, \text{optional} \diamond \psi) \wedge \\ \text{done}(\text{communicate}(a, b, \neg \text{BEL}(a, \text{optional} \diamond \psi))) \wedge \\ \text{done}(\text{coordinate}(a, b, \psi)))\}] \end{aligned} \quad (4.2.1.II)$$

- la stratégie d'**engagement ouvert** (*open-minded*), l'agent a abandonne son engagement lorsque cet engagement ne correspond plus à son but :

$$\begin{aligned} \text{COMM}(a, b, \psi) \rightarrow \\ \text{inevitable}[\text{COMM}(a, b, \psi)U \\ \{\text{BEL}(a, \psi) \vee \\ (\neg \text{BEL}(a, \text{optional} \diamond \psi) \wedge \\ \text{done}(\text{communicate}(a, b, \neg \text{BEL}(a, \text{optional} \diamond \psi))) \wedge \\ \text{done}(\text{coordinate}(a, b, \psi))) \\ \vee (\neg \text{GOAL}(a, \psi) \wedge \\ \text{done}(\text{communicate}(a, b, \neg \text{GOAL}(a, \psi))) \wedge \\ \text{done}(\text{coordinate}(a, b, \psi)))\}] \end{aligned} \quad (4.2.1.III)$$

Plus la stratégie est forte, plus l'intention est persistante. La stratégie d'engagement aveugle (blind) est plus forte que la stratégie d'engagement simple (single-minded) qui est plus forte que la stratégie d'engagement ouvert (open-minded).

En résumé, le cadre défini dans [20] permet de résoudre collaborativement un problème de manière flexible. À cet effet, les agents sont impliqués dans des dialogues de demande d'information, dans des dialogues de persuasion et dans des dialogues de négociation. Le but de ces dialogues consiste à établir des engagements sociaux selon leur stratégie individuelle d'engagement. La section suivante décrit comment des dialogues entre agents peuvent se dérouler.

4.2.2 DIAGAL

Suite aux travaux de Nicolas Maudet [39, 40] et de Brahim Chaib-Draa [17] concernant les dialogues dans les SMA, Philippe Pasquier, Mathieu Bergeron et Brahim Chaib-draa proposent un langage de communication d'agents intitulé DIAGAL [47, 37, 48] (*DIALOGue-Game based Agent Language*). Fors des critiques émises sur l'approche mentalistique de la communication inter-agents, ce langage est adapté à un SMA ouvert et hétérogène.

Nous allons dans un premier temps présenter la formalisation de la notion d'engagement social telle qu'elle a été proposée (cf section 4.2.2.1). Dans un second temps, nous verrons comment cette notion d'engagement social est utilisée par les dialogues (cf section 4.2.2.2).

4.2.2.1 Engagement social

La notion d'engagement social est ici cruciale. Comme dans la section précédente, la notion d'engagement social peut être formalisée à l'aide d'un prédicat.

Définition n° 4.2.2.I [Engagement social]

On appelle **engagement social** le prédicat suivant :

$$C(x, y, \alpha, t, s_x, s_y) \quad (4.2.2.I)$$

D'après ce prédicat, l'agent x , appelé **débiteur**, s'engage socialement vis à vis de l'agent y , appelé **crédeur**, à réaliser l'action α , appelée **contenu**, au temps t . Si l'agent x décide d'annuler l'engagement il se verra infliger la **sanction** s_x alors que l'agent y qui libère l'engagement se verra infliger la **sanction** s_y .

Il est important de remarquer que le contenu d'un engagement n'est pas nécessairement une action atomique. Le contenu peut être un choix non déterministe (par exemple, $\alpha_1|\alpha_2$) ou une action conditionnée par une autre action (par exemple, $\alpha_1 \rightarrow \alpha_2$). De plus, le contenu d'un engagement social peut être une formule logique (notée ϕ, ψ, \dots). On parle alors d'**engagement propositionnel**.

La notion d'engagement social est dynamique. Un engagement est créé puis il évolue. L'état d'un engagement est stocké dans des **agendas**, *i.e.* des structures de données distribuées parmi les agents. Un engagement qui n'est pas socialement accepté est **inactif** (noté **Inactive**). Il est **créé** (noté **Created**) s'il est socialement établi entre un crédeur et un débiteur. Dans le cas contraire, un engagement qui n'a pu être établi socialement est dans un état d'**échec** (noté **Failed**). On dit d'un engagement

qu'il est **annulé** (noté *Canceled*) lorsqu'il est retiré par son crédeur. À l'inverse, on dit d'un engagement qu'il est **libéré** (noté *Released*) lorsqu'il est retiré par son débiteur. Un engagement est **rempli** (noté *Full*) dans la mesure où le débiteur a respecté son contenu. Un engagement est **violé** (noté *Violated*) si c'est le crédeur qui n'a pas respecté son contenu.

La figure 4.2.2.I représente le cycle de vie d'un engagement sous la forme d'un automate fini déterministe (cf définition 1.4.2.II). L'état *Inactive* est initial. Les états *Failed*, *Canceled*, *Released*, *Full*, *Violated* sont finaux.

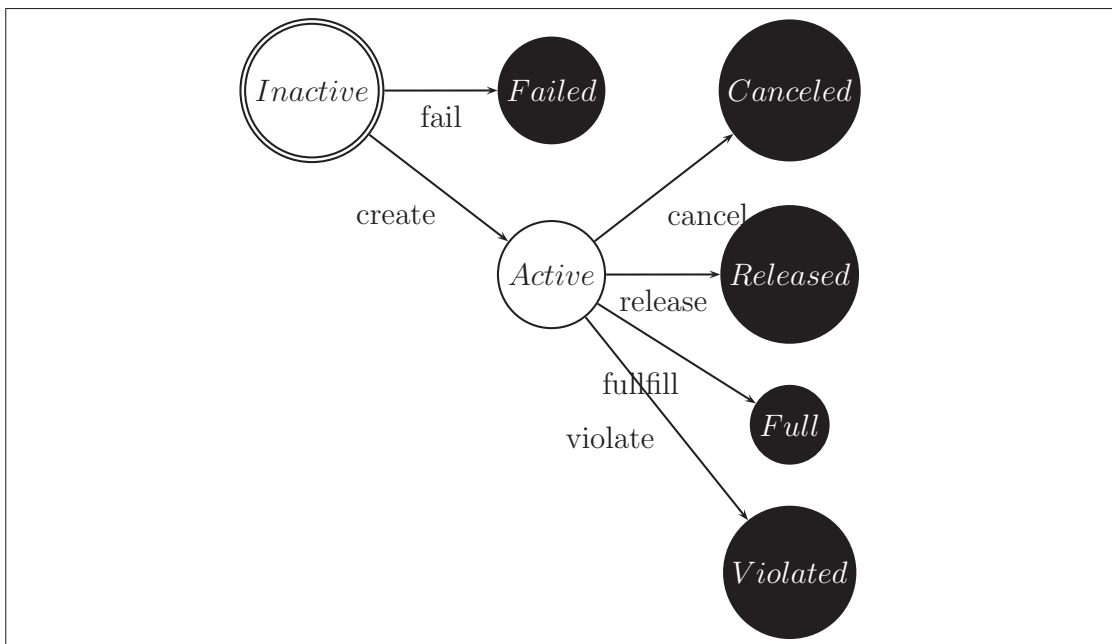


Fig. 4.2.2.I: Cycle de vie d'un engagement

Les transitions correspondent à trois types d'évènements qui peuvent modifier l'état de l'engagement : la réalisation d'une action (non communicative), l'écoulement du temps ou la réalisation d'un jeu de dialogue. Nous nous intéresserons ici uniquement au jeu de dialogue.

4.2.2.2 Jeu de dialogue

Le langage DIAGAL s'appuie sur la notion de jeu de dialogue. Il est constitué d'un ensemble de 12 jeux élémentaires (*basic game*) qui peuvent être combinés.

Les **jeux de dialogue** sont des structures collaboratives de communication. Elles permettent de coordonner l'activité dialogique des agents grâce aux engagements (C) et aux méta-engagements (C_g) créés. D'une part les conditions sont des engagements qui déterminent quand le jeu est initié ou quand il se termine. Pour entrer dans un

jeu de dialogue, les conditions d'entrée d'un jeu (notées E) doivent être remplies. Les conditions de succès d'un jeu (notées S) représentent l'engagement final désiré, *i.e.* le but de l'agent initiateur. Les conditions d'échec d'un jeu (notées F) permettent de déduire que le jeu a subi un échec. L'engagement désiré n'est pas rempli. D'autre part, les règles dialectiques d'un jeu (notées R) sont définies en terme de méta-engagements qui doivent être respectés par les joueurs. Ces règles permettent au dialogue de progresser.

L'exemple 4.2.2.I permet d'illustrer cette structuration au travers du jeu de requête.

Exemple n° 4.2.2.I [Le jeu de requête] *Ce jeu (noté rg) permet à un agent x d'envoyer une requête à un agent y en vue de l'engager à réaliser une action α . L'agent y qui reçoit la requête peut soit accepter de réaliser l'action soit refuser. En cas de succès un engagement concernant l'action α sera créé. Dans le cas contraire, le jeu se termine sur un échec. Les règles et les conditions de ce jeu sont présentées dans le tableau 4.2.2.I.*

E_{rg}	$\neg C(y, x, \alpha, t_i)$ et $\neg C(y, x, \neg\alpha, t_i)$
S_{rg}	$C(y, x, \alpha, t_f)$
F_{rg}	$\neg C(y, x, \alpha, t_f)$
R_{rg}	<ol style="list-style-type: none"> 1) $C_g(x, y, \text{request}(x, y, \alpha), t_j)$ 2) $C_g(y, x, \text{request}(x, y, \alpha) \rightarrow$ $C_g(y, x, \text{accept}(y, x, \alpha) \text{refuse}(y, x, \alpha), t_{j+1}), t_j)$ 3) $C_g(y, x, \text{accept}(y, x, \alpha) \rightarrow C(y, x, \alpha, t_f), t_j)$ 4) $C_g(y, x, \text{refuse}(y, x, \alpha) \rightarrow \neg C(y, x, \alpha, t_f), t_j)$

Tab. 4.2.2.I: Conditions et règles du jeu de requête

Le tableau 4.2.2.II présente les différents jeux de dialogue. Dans ce tableau, on distingue dans la partie supérieure les jeux qui portent sur une action et dans la partie inférieure les jeux qui portent sur une formule. On distingue d'une part les jeux qui portent sur un engagement dont l'initiateur est le créateur et le partenaire est le débiteur et d'autre part les jeux qui portent sur un engagement dont l'initiateur est le débiteur et le partenaire est le créateur. Un jeu peut avoir pour but d'établir ou de supprimer un engagement social.

En résumé, la notion d'engagement social est utilisée d'une part pour gérer le dialogue. On parle d'**engagement dialogique**. Il capture l'aspect conventionnel de la pragmatique. D'autre part, le but d'un dialogue consiste à établir un engagement. On parle alors d'**engagement extra-dialogique**.

Dans la section suivante, nous présentons un modèle de dialogue entre agents qui s'appuie moins sur la notion d'engagement que sur celle d'argument.

Type d'engagement	But du dialogue	jeu de dialogue
$co = C(x, y, \alpha)$	co	Jeu d'offre
	$\neg co$	Jeu d'annulation d'action
$co = C(y, x, \alpha)$	co	Jeu de requête
	$\neg co$	Jeu de libération d'action
$co = C(x, y, \phi)$	co	Jeu d'information
	$\neg co$	Jeu d'annulation de proposition
$co = C(y, x, \phi)$	co	Jeu de question
	$\neg co$	Jeu de libération de proposition

Tab. 4.2.2.II: Classification des jeux de dialogue utilisés par DIAGAL

4.3 Dialogue et arguments

Un système multi-agents argumentatifs est consisté d'un ensemble d'agents autonomes et sociaux qui argumentent (cf section 4.3.1) et dans lequel on peut circonscrire un cadre, appelé système dialectique multi-agents, au travers duquel deux agents dialoguent (cf section 4.3.2).

4.3.1 Système multi-agents argumentatifs

Le modèle AMP¹ [4, 46, 5, 3] a été proposé par Leila Amgoud, Nicolas Maudet et Simon Parsons. D'après ce modèle, un système multi-agents argumentatifs est un ensemble d'agents munis de leurs propres arguments qui ne sont pas connus *a priori* par leur interlocuteur. Les agents utilisent le modèle de raisonnement présenté dans la section 2.4.2. Ils dialoguent à l'aide d'un langage de communication d'agents. Ce langage est muni d'une sémantique argumentative et d'une sémantique sociale.

Un **système multi-agents argumentatifs** est constitué de deux joueurs : un agent validateur ag_V et un agent falsificateur ag_F . L'agent validateur (respectivement l'agent falsificateur) est muni de sa propre base de connaissances, notée Σ_V (respectivement Σ_F). De plus, un agent est muni d'un tableau d'engagements qui répertorie les déclarations émises par son interlocuteur au cours du dialogue.

Plus formellement, chacun des agents est conforme à la définition suivante :

¹du nom de ses auteurs Amgoud/Maudet/Parsons.

Définition n° 4.3.1.I [Agent argumentatif]

Un **agent argumentatif** $ag_i \in \{ag_P, ag_F\}$ est défini par un couple $ag_i = \langle \Sigma_i, CS_j \rangle$ où :

- Σ_i est la base de connaissances de l'agent, i.e. un ensemble d'expressions bien formées qui constituent les croyances de l'agent ;
- CS_j est le tableau d'engagement constitué des engagements pris par son interlocuteur (ag_j avec $ag_j \neq ag_i$) ;

Le tableau d'engagement, noté CS_j , répertorie l'ensemble des expressions bien formées sur lesquelles l'agent ag_j s'engage. Initialement vides, les tableaux d'engagement conservent la trace des engagements au cours des échanges.

Parce que chaque agent ag_i dispose d'une base de connaissances (Σ_i) munie d'une relation de priorité (\ll_i), il est associé au système d'argumentation à base de préférences (cf définition 2.4.2.V) suivant :

$$PAS_i = \langle \mathcal{A}(\Sigma_i), \text{undercut}, \text{pref}_i \rangle$$

L'ensemble des arguments acceptables associé est noté \mathcal{S}_i . Contrairement à un jeu argumentatif d'évaluation sémantique (cf section 3.5), chacun des joueurs est muni de son propre argumentaire $\mathcal{A}(\Sigma_i)$ qui s'appuie sur ses seules croyances.

Les agents échangent des connaissances, i.e. expressions bien formées. Ainsi, ils partagent implicitement un même langage de représentation des connaissances (noté L).

Pour échanger des connaissances, un langage de communication d'agents est nécessaire (noté \mathcal{CL}). Chaque message est muni d'un identifiant unique $M_k \in \mathcal{CL}$. Il contient un acte de langage ($A_k = \text{act}(M_k)$). Cet acte de langage est constitué d'une locution et d'un contenu propositionnel. Le contenu propositionnel, appelé hypothèse, est constitué d'une expression bien formée ($\psi \in \mathcal{L}$) ou d'un ensemble d'expressions bien formées ($\Phi \subset \mathcal{L}$). La locution peut être une affirmation (assert), une mise en doute (challenge), une question (question), une acceptation (accept) ou un avis d'ignorance (unknow). Ce langage de communication d'agents est muni d'une sémantique argumentative et d'une sémantique sociale.

La **sémantique sociale** des actes permet de les interpréter. Selon l'acte reçu, les tableaux d'engagement sont (ou non) mis à jour selon les règles suivantes :

Définition n° 4.3.1.II [Règle de mise à jour des engagements]

Soit $M_{k+1} \in \mathcal{CL}$, un message émis par l'agent ag_j .

Si $act(M_{k+1}) = question(H)$ ou $act(M_{k+1}) = unknow(H)$ ou $act(M_{k+1}) = challenge(H)$ alors $CS_j(M_{k+1}) = CS_j(M_k)$.

Si $act(M_{k+1}) = assert(H)$ ou $act(M_{k+1}) = accept(H)$ alors $CS_j(M_{k+1}) = CS_j(M_k) \cup H$.

Les agents prennent note des hypothèses émises par leurs interlocuteurs.

Alors que la sémantique sociale du langage de communication permet d'interpréter un acte, la **sémantique argumentative** du langage de communication détermine dans quelle mesure un acte peut être énoncé. C'est à partir du système d'argumentation associé à l'agent que sont spécifiées les conditions rationnelles d'énonciation des actes de langage. La condition rationnelle d'énonciation d'un acte de langage est fonction du contenu propositionnel et dépend également de la locution utilisée.

Un agent peut affirmer une hypothèse dans la mesure où il possède un argument en sa faveur.

Définition n° 4.3.1.III [Condition d'énonciation d'une affirmation]

On note $can_assert(ag_i, H)$ le prédicat défini tel que :

$$can_assert(ag_i, H) \Leftrightarrow \forall h \in H \exists A \in \mathcal{A}(\Sigma_i) \text{ conclusion}(A) = h$$

$can_assert(ag_i, H)$ est appelé **condition rationnelle d'énonciation de l'affirmation de l'hypothèse H par un agent ag_i** .

Dans le modèle AMP, la condition rationnelle d'énonciation d'une affirmation et celle d'une acceptation sont identiques.

Définition n° 4.3.1.IV [Condition d'énonciation d'une acceptation]

On note $can_concede(ag_i, H)$ le prédicat défini tel que :

$$can_concede(ag_i, H) \Leftrightarrow \forall h \in H \exists A \in \mathcal{A}(\Sigma_i) \text{ conclusion}(A) = h$$

$can_concede(ag_i, H)$ est appelé **condition rationnelle d'énonciation de l'acceptation de l'hypothèse H par un agent ag_i** .

Les autres actes de langage ($question(H)$, $challenge(H)$, $unknow(H)$) n'ont pas de conditions rationnelles d'énonciation particulières.

Les conditions rationnelles d'énonciation sont des prédicats dont les définitions ne diffèrent pas d'un agent à un autre. Ces conditions ne sont pas nécessairement mutuellement exclusives. Un agent peut affirmer une hypothèse et son contraire. L'agent doit effectuer un choix. C'est l'objet des tactiques d'énonciation. Elles dépendent de son attitude.

Un agent prévenant souhaite affirmer une hypothèse dans la mesure où il dispose d'un argument acceptable en sa faveur. Sans cette restriction l'agent est confiant.

De manière similaire, un agent sceptique souhaite accepter une hypothèse dans la mesure où il dispose d'un argument acceptable en sa faveur. Sans cette restriction l'agent est crédule.

Dans le modèle AMP, les agents argumentent afin d'échanger leurs connaissances. À cet effet, il utilise un langage de communication d'agents muni d'une sémantique argumentative et d'une sémantique sociale. Bien que la problématique liée à l'enchaînement des messages soit abordée dans le modèle AMP, les protocoles ne sont pas complètement formalisés. De même, le cadre permettant d'évaluer leurs propriétés reste implicite. C'est l'objet du travail présenté dans la section suivante.

4.3.2 Système dialectique multi-agents

Henry Prakken [50, 49] utilise la théorie de la preuve dialectique présentée dans la section 3.5 pour modéliser des persuasions dans lesquelles deux agents s'affrontent. Il introduit la notion de système dialectique multi-agents. Un tel système est constitué de deux agents munis de leurs propres arguments, comme dans le modèle AMP. De plus, l'objectif de ce travail consiste à proposer un certain nombre de protocoles de discussion et à les évaluer.

Nous allons dans un premier temps présenter la notion de système dialectique multi-agents (cf section 4.3.2.1) pour voir dans quelle mesure elle intègre la notion de protocoles de discussion (cf section 4.3.2.2). Afin de proposer un protocole de discussion complet et correct, nous allons envisager une première solution dite naïve dans la section 4.3.2.3. Comme ce type de protocole n'est pas satisfaisant, nous envisagerons dans la section 4.3.2.4 une seconde solution, moins naïve.

4.3.2.1 Cadre

Dans la théorie de la preuve dialectique (cf section 3.5), le système dialectique s'appuie sur un système d'argumentation abstrait. En d'autres termes les arguments sont considérés comme des entités abstraites (cf section 2.3). Au contraire, un système multi-agents dialectique s'appuie sur une logique argumentative (cf section

2.4). Ainsi, on distingue l'ensemble des arguments qui peuvent être construits à partir du langage logique sous jacent ($\text{Arg}(\mathcal{L})$) d'une part, et l'ensemble des arguments qui peuvent être extraits d'une théorie ($\text{Arg}(\mathcal{T})$), *i.e.* à partir d'un ensemble d'expressions bien formées ($\mathcal{T} \subset \mathcal{L}$). Ne plus considérer les arguments comme des entités abstraites permet de spécifier les actes de langage qui sont échangés.

Comme dans la section précédente, l'acte de langage associé à un message ($\text{act}(M_n)$) est constitué d'une locution et d'un contenu propositionnel. Toutefois, la liste des locutions diffère légèrement. Un acte de langage peut être une affirmation (*assert*), une mise en doute (*challenge*), une concession (*concede*), une justification (*argue*) ou une rétractation (*retract*). Un acte de langage (act_B) permet de répondre à l'un des actes précédemment émis (act_A). Il permet d'attaquer ($\text{act}_B = \text{attacks}(\text{act}_A)$) ou d'abdiquer ($\text{act}_B = \text{surrenders}(\text{act}_A)$) face à l'acte de langage auquel il fait référence.

Dans le tableau 4.3.2.I, l'ensemble des actes de langage sont répertoriés. Une mise en doute attaque une précédente affirmation alors qu'une concession abdique face à cette affirmation. Une justification attaque une mise en doute alors qu'une rétractation abdique face à cette mise en doute. Une mise en doute ou un contre-argument attaque une précédente justification alors qu'une concession abdique face à cette justification. Une concession qui répond à une justification peut porter soit sur l'une des formules du support de l'argument précédemment avancé soit sur la totalité de cet argument. Contrairement aux actes de langage qui attaquent l'un des actes précédents, les actes de langage qui abdiquent ne peuvent donner lieu à aucune réponse.

Actes de langage	Attaque	Abdication
$\text{assert}(\psi)$	$\text{challenge}(\psi)$	$\text{concede}(\psi)$
$\text{challenge}(\psi)$	$\text{argue}(\Phi \rightarrow \psi)$	$\text{retract}(\psi)$
$\text{concede}(\psi)$	\emptyset	\emptyset
$\text{retract}(\psi)$	\emptyset	\emptyset
$\text{argue}(\Phi \rightarrow \psi)$	$\text{challenge}(\phi \in \Phi)$ $\text{argue}(\Phi' \rightarrow \psi')$	$\text{concede}(\phi \in \Phi)$ $\text{concede}(\Phi \rightarrow \psi)$
$\text{concede}(\Phi \rightarrow \psi)$	\emptyset	\emptyset

Tab. 4.3.2.I: Ensemble des actes de langage et leurs réponses potentielles

Comme dans la section précédente, les tableaux d'engagement sont (ou non) mis à jour selon les règles suivantes :

Définition n° 4.3.2.I [Règles de mise à jour des engagements]

Dans un système dialectique, une énonciation M_n modifie le tableau d'engagements de l'émetteur $CS_{player(M_n)}(h_n)$ selon les **règles d'engagement** suivantes :

- si $act(M_n) = assert(\psi)$ ou $act(M_n) = concede(\psi)$ alors $CS_{player(M_n)}(h_n) = CS_{player(M_n)}(h_{n-1}) \cup \{\psi\}$;
- si $act(M_n) = argue(\Phi \rightarrow \psi)$ ou $act(M_n) = concede(\Phi \rightarrow \psi)$ alors $CS_{player(M_n)}(h_n) = CS_{player(M_n)}(h_{n-1}) \cup \Phi \cup \{\psi\}$;
- si $act(M_n) = retract(\psi)$ alors $CS_{player(M_n)}(h_n) = CS_{player(M_n)}(h_{n-1}) - \{\psi\}$;
- si $act(M_n) = challenge(\psi)$ alors $CS_{player(M_n)}(h_n) = CS_{player(M_n)}(h_{n-1})$.

Étant donné la nouvelle liste de locutions, la définition 4.3.1.II a été modifiée. Cette nouvelle définition est conforme aux notations utilisées dans la section 3.5.

Un **système dialectique multi-agents** est un dispositif à travers lequel deux agents s'affrontent à propos de la validité d'une formule ψ_0 . L'agent validateur affirme que ψ_0 est vraie et l'agent falsificateur affirme le contraire. Ils s'appuient tous deux sur le même système d'argumentation mais chacun est muni de son propre argumentaire. La définition suivante décrit plus formellement un système dialectique :

Définition n° 4.3.2.II [Système dialectique multi-agents]

Soient $AS = \langle \text{Arg}(\mathcal{L}), \text{attacks} \rangle$ un système d'argumentation, et $\psi_0 \in \mathcal{L}$ une formule du langage sous-jacent. Un **système dialectique multi-agents** qui porte sur ψ_0 est un tuple $DS_{\text{Moves}}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, \text{convention}, Z, (u_p)_{p \in N} \rangle$ défini tq :

- $N = \{ag_V, ag_F\}$ est un ensemble de deux joueurs : un agent valideur et un agent falsificateur, chacun muni de sa propre théorie : \mathcal{T}_V et \mathcal{T}_F ;
- Moves est un ensemble de coups dialogiques bien formés. Un coup dialogique est soit un coup d'initialisation : $M_k = \langle \text{player}, \text{act}_k \rangle$ soit un coup de réponse : $M_k = \langle \text{player}, \text{act}_k, M_i \rangle$ tq le premier élément (noté $\text{player}(M_k)$) est un joueur, le second élément (noté $\text{act}(M_k)$) est un acte de langage et s'il y a lieu le troisième élément (noté $\text{move}(M_k)$) est un coup. Si $\text{move}(M_k) = M_i$, on dit que M_k répond à M_i ;
- H est l'ensemble des historiques, i.e. des séquences finies de coups $h_n = (M_1, \dots, M_n)$ définis tq $\forall 1 \leq k \leq n$:
 1. le joueur d'un coup est déterminé par une fonction de tour de parole : $\text{player}(M_k) = T(M_1, \dots, M_{k-1})$;
 2. les coups modifient le tableau d'engagements $CS_{\text{player}(M_k)}(h_k)$;
 3. les coups doivent être conformes à une convention : $M_k \in \text{convention}(M_1, \dots, M_{k-1})$.
- $T : H \rightarrow N$ est une fonction de tour de parole qui détermine, étant donné un historique, le joueur du coup suivant. Si l'historique est de longueur nulle ou paire alors $T(h) = ag_V$ sinon $T(h) = ag_F$;
- $CS : H \times N \rightarrow 2^{\mathcal{L}}$ est une fonction qui associe à un historique et à un participant, un ensemble de formules sur lesquelles le participant s'est engagé au cours du dialogue. Les tableaux d'engagements sont initialement vides ($CS_V(\emptyset) = CS_F(\emptyset) = \emptyset$) ;
- $\text{convention} : H \times CS_V \times CS_F \rightarrow \text{Moves}$ est une fonction qui, étant donné un historique et l'état des tableaux d'engagements, détermine l'ensemble des coups autorisés ;
- Z est l'ensemble des dialogues, i.e. les historiques de tailles maximales ;
- $u_V, u_F : H \rightarrow \{-1, 1\}$, i.e. les gains sont des fonctions éventuellement partielles qui déterminent le vainqueur d'un historique.

Contrairement à un système dialectique, les participants sont des agents munis de leurs propres arguments : \mathcal{T}_V et \mathcal{T}_F . Les coups ne contiennent plus simplement des arguments abstraits mais des actes de langage munis d'un contenu propositionnel. De plus, les tableaux d'engagements ont été adjoints afin de notifier les formules

sur lesquelles les participants s'engagent au cours d'un dialogue. Ainsi, la convention dépend non seulement de l'historique mais également de l'état des tableaux d'engagements. Le protocole de discussion spécifie la convention à respecter.

4.3.2.2 Protocole de discussion

Un **protocole de discussion** (que nous appellerons simplement protocole) est d'une part constitué d'une convention, *i.e.* un ensemble de règles dialectiques qui spécifient les messages autorisés (ou non). D'autre part, il permet d'attribuer la victoire sur un historique à l'un des participants. C'est l'objet des gains $((u_p)_{p \in N})$. La définition suivante décrit plus formellement un protocole :

Définition n° 4.3.2.III [Protocole de discussion]

Soit $DS_{Moves}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur ψ_0 . Le couple $Proto = \langle convention, (u_p)_{p \in N} \rangle$ est appelé **protocole de discussion** ssi :

1. la fonction convention vérifie que :

$$(a) \text{ convention}(\emptyset) = \langle V, \text{assert}(\psi_0) \rangle ;$$

$$(b) \text{ i) } M_k \in \text{convention}(M_1, \dots, M_{k-1}) \text{ (avec } k > 1) \text{ si } \exists i, 1 \leq i < k \text{ tq :}$$

$$\begin{aligned} \text{move}(M_k) = M_i \wedge \text{player}(M_k) \neq \text{player}(M_i) \wedge \\ (\text{act}(M_k) = \text{surrenders}(\text{act}(M_i)) \vee \text{act}(M_k) = \text{attacks}(\text{act}(M_i))) \end{aligned}$$

$$\text{ii) Si } \exists j, i < j < k \text{ tq } \text{move}(M_j) = M_i \text{ alors } \text{act}(M_k) \neq \text{act}(M_j) ;$$

$$(c) M_k \in \text{convention}(h_{k-1}) \text{ (avec } k > 1) \text{ si } CS_{\text{player}(M_k)}(h_k) \not\vdash \perp ;$$

$$(d) \text{ Si } CS_{\text{player}(M_k)}(h_{k-1}) \not\vdash \psi \text{ et } CS_{\text{player}(M_k)}(h_{k-1}) \not\vdash \neg \psi \text{ alors} \\ \text{act}(M_k) = \text{concede}(\psi) \text{ est autorisé ;}$$

$$(e) \text{ Si } \psi \notin CS_{\text{player}(M_k)}(h_{k-1}) \text{ alors } \text{act}(M_k) = \text{assert}(\psi) \text{ est autorisé ;}$$

$$(f) \text{ Si } \psi \in CS_{\text{player}(M_k)}(h_{k-1}) \text{ alors } \text{act}(M_k) = \text{retract}(\psi) \text{ est autorisé ;}$$

$$(g) \text{ Si } CS_{\text{player}(M_k)}(h_{k-1}) \not\vdash \psi \text{ alors } \text{act}(M_k) = \text{challenge}(\psi) \text{ est autorisé ;}$$

2. Les gains vérifient que :

$$(a) u_F = -u_V ;$$

$$(b) \text{ convention}(h) = \emptyset \rightarrow$$

$$((T(h) = \text{ag}_V \rightarrow u_V(h) = -1) \wedge (T(h) = \text{ag}_F \rightarrow u_V(h) = 1))$$

La fonction convention qui spécifie les coups dialogiques autorisés (ou non) vérifie les propriétés suivantes :

- (a) Le seul coup d'initialisation est une affirmation de ψ_0 par l'agent validateur.
- (b) Les coups de réponse sont autorisés dans la mesure où :
 - i) ils répondent à un des coups précédents de son adversaire. L'acte de langage doit attaquer ou abdiquer face à l'acte de langage du coup auquel il répond ;
 - ii) deux coups qui répondent au même coup ne peuvent pas contenir le même acte de langage.
- (c) Un coup est autorisé dans la mesure où il conserve la consistance du tableau d'engagements.
- (d) Une formule peut être concédée si elle ou sa négation ne peut être déduite.
- (e) Une formule peut être affirmée si elle n'a pas été précédemment affirmée.
- (f) Une formule peut être rétractée si elle a été précédemment affirmée.
- (g) Une formule peut être mise en doute si rien ne la justifie.

Les gains $(u_p)_{p \in N}$, qui déterminent le vainqueur d'un historique, vérifient les propriétés suivantes :

- (a) Si un joueur gagne, l'autre joueur perd.
- (b) Le joueur qui a le dernier mot gagne.

Les retours en arrière (*backtracks*) sont *ipso facto* autorisés. La ligne persuasive d'un historique est la sous-séquence dans laquelle ces retours en arrière ont été évincés.

Définition n° 4.3.2.IV [Ligne persuasive]

Soient $DS_{Moves}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur ψ_0 et $h_k = (M_1, \dots, M_k) \in H$ un historique. La **ligne persuasive** de h est la plus petite sous-séquence de h_k (noté $line(h_k)$) tq $M_1, M_k \in line(h_k) \wedge \forall M_i \in line(h_k) M_i \in convention(M_1, \dots, M_{i-1})$.

Un **protocole** peut être à **coups uniques** ou à **coups multiples**. Dans le premier cas, un joueur peut émettre un seul coup par tour de parole. Dans le second cas, il peut émettre plusieurs coups par tour de parole. Nous n'envisagerons ici que le premier cas. Un **protocole** peut être à **réponses uniques** ou à **réponses multiples**. Dans le premier cas, un joueur ne peut apporter qu'une seule réponse à chacun des coups de son adversaire. Les historiques sont tous des lignes persuasives. Dans le second cas, un joueur peut essayer plusieurs actes de langage pour attaquer ceux de son adversaire.

Définition n° 4.3.2.V [Protocole (à réponses uniques/multiples)]

Soit $DS_{\text{Moves}}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents. Soit $\text{Proto} = \langle \text{convention}, (u_p)_{p \in N} \rangle$ un protocole de discussion.

– *Proto* est un **protocole à réponses uniques** ssi

$$\forall M_k \in \text{convention}(h_{k-1}) \text{ (avec } k > 1), \text{move}(M_k) = M_{k-1};$$

– *Proto* est un **protocole à réponses multiples** ssi

$$\forall M_k \in \text{convention}(h_{k-1}) \text{ (avec } k > 1) \exists M_i \in \text{line}(h_{k-1}), \text{move}(M_k) = M_i.$$

Dans un protocole à réponses uniques, les retours arrières sont interdits. Dans un protocole à réponses multiples, les retours arrières sont autorisés à la condition qu'ils fassent référence à un des coups de la même ligne persuasive. Ces retours arrière ne peuvent pas faire référence à une branche précédemment explorée puis abandonnée. Cette condition a pour but de réduire l'espace de recherche dans lequel se situent les coups autorisés. Nous verrons par la suite que cette hypothèse est restrictive.

Dans quelle mesure un protocole à réponses uniques est-il plus (ou moins) efficace qu'un protocole à réponses multiples? Tout dépend du contexte². Quand on désire une décision rapide, un protocole à réponses uniques est préférable. On force chacun des joueurs à avancer ses arguments les plus forts sans perdre de temps sur des choix moins prometteurs. Quand la qualité du résultat prévaut, un protocole à réponses multiples est plus approprié. Tous les arguments à la disposition des joueurs peuvent être essayés. L'efficacité des dialogues dépend non seulement du protocole mais également de la stratégie des participants.

Comme dans un jeu argumentatif d'évaluation sémantique, une stratégie désigne un ensemble de choix d'action décidés *a priori*. La définition 3.5.0.XVI a été modifiée de la manière suivante :

²Cette réponse peut être qualifiée de normande. Soyons plus précis.

Définition n° 4.3.2.VI [Stratégie (déterminée)]

Soit $DS_{Movers}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur ψ_0 . $S_V \subseteq H$ est **une stratégie (déterminée)** pour l'agent valideur ssi il existe une fonction :

$$f_V : T^{-1}(\{V\}) \cap S_V \rightarrow Moves$$

tel que S_V est le plus petit ensemble qui satisfasse les conditions suivantes :

1. si $T(M_1, \dots, M_{k-1}) = ag_F$ alors $(M_1, \dots, M_k) \in S_V$;
2. si $T(M_1, \dots, M_{k-1}) = ag_V$ alors $(M_1, \dots, M_{k-1}, f_V(M_1, \dots, M_{k-1})) \in S_V$, la fonction f_V sélectionnant le coup de réponse M_k pour étendre l'historique.

Une stratégie pour l'agent falsificateur peut être définie de manière similaire. Une stratégie gagnante pour l'agent valideur est une stratégie qui mène à un historique qu'il gagne.

Définition n° 4.3.2.VII [Stratégie gagnante]

Soit $DS_{Movers}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur ψ_0 . Soit S_V une stratégie pour l'agent valideur. S_V est une **stratégie gagnante** pour l'agent valideur ssi

$$\forall h_n \in S_V \cap H \quad u_V(h_n) = 1.$$

Une stratégie gagnante pour l'agent falsificateur peut être définie de manière similaire.

Une formule est prouvable avec crédulité ssi l'agent valideur a une stratégie gagnante dans le système dialectique correspondant. Dans le cas contraire, la formule n'est pas prouvable avec crédulité.

Définition n° 4.3.2.VIII [Prouvabilité crédule]

Soit $DS_{Movers}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur ψ_0 . ψ_0 est **prouvable avec crédulité** dans AS (noté $provable^{AS}(\psi_0)$) ssi il existe une stratégie gagnante pour l'agent valideur. Une formule n'est pas prouvable dans AS (noté $\neg provable^{AS}(\psi_0)$) ssi il existe une stratégie gagnante pour l'agent falsificateur.

Un tel jeu est déterminé, *i.e.* il existe une stratégie gagnante pour exactement un des joueurs. En effet, c'est un jeu à information parfaite, à somme nulle, pour deux

joueurs, avec deux résultats de jeu et un nombre fini d'actions. D'après le théorème de Zermelo, ce type de jeu est déterminé [65].

Dans un jeu argumentatif d'évaluation sémantique, l'ensemble des arguments sur lequel s'appuie le système dialectique représente l'ensemble des arguments qui peuvent être échangés. C'est à partir de cette base d'arguments que la victoire est attribuée à l'un des joueurs. En d'autres termes, le joueur qui a le dernier mot dans la base d'arguments gagne. Les gains sont ainsi déterminés. Il est important de noter que cette base d'arguments est fixée et connue *a priori*.

Dans une discussion critique, ce n'est pas le cas. L'ensemble des arguments sur lequel s'appuie un système dialectique multi-agents est considéré comme un langage d'arguments bien formés. C'est à partir de ce langage que sont élaborés les arguments qui peuvent être avancés, voire rétractés. L'ensemble des arguments qui peuvent être échangés constitue la base d'arguments. C'est à partir de cette base d'arguments que la victoire est attribuée à l'un des agents. En d'autres termes, l'agent qui a le dernier mot dans la base d'arguments gagne. Les gains sont ainsi déterminés. Il est important de noter que cette base d'arguments est dynamique.

La base d'arguments d'une discussion critique est constituée de l'ensemble des arguments notifiés dans les tableaux d'engagements. On note $\text{Arg}(\text{CS}_V(h) \cup \text{CS}_F(h))$ l'ensemble des arguments émis au cours de l'historique h . De plus, une discussion peut débiter sur un accord partiel. Les joueurs peuvent admettre conjointement un ensemble initial d'arguments. On le note $\text{Arg}(\mathcal{T}_{\text{init}})$. Selon la logique sous-jacente sur laquelle s'appuie la structure des arguments, ces ensembles d'arguments peuvent contenir des arguments « implicites » : soit des sous-arguments ; soit d'autres arguments qui peuvent être inférés. Ces arguments supplémentaires sont contenus dans la clôture de ces ensembles.

Définition n° 4.3.2.IX [Clôture d'un ensemble d'arguments]

Soient $\text{Arg}(\mathcal{L})$ un langage d'arguments bien formés et $S, S' \subset \text{Arg}(\mathcal{L})$ deux ensembles d'arguments. La clôture de S (noté $Cl(S)$) est définie tq :

1. $S \subseteq Cl(S)$;
2. si $S' \supseteq S$ alors $Cl(S') \supseteq Cl(S)$;
3. $Cl(S) \subseteq \text{Arg}(\mathcal{L})$.

On dit que $Cl(S)$:

- n'est **pas calculée** ssi $Cl(S) = S$;
- est **partiellement calculée** ssi $Cl(Cl(S)) \subset Cl(S)$;
- est **totalelement calculée** ssi $Cl(Cl(S)) = Cl(S)$.

La clôture d'un ensemble d'arguments est monotone (cf condition 2 de la définition

4.3.2.IX). Elle peut être obtenue sans inférence, on dit qu'elle n'est pas calculée. La clôture peut être obtenue par l'inférence d'un agent dont la rationalité est limitée, on dit alors que la clôture est partiellement calculée. La clôture peut être obtenue par inférence d'un agent dont la rationalité n'est pas limitée, on dit alors que la clôture est totalement calculée.

En conséquence, la base des arguments est dynamique. Elle dépend de l'historique envisagé et des inférences qui peuvent être effectuées à partir des arguments échangés.

Définition n° 4.3.2.X [Base d'arguments]

Soit $DS_{Moves}(\psi_0, Arg(\mathcal{L})) = \langle N, H, T, (CS_p)_{p \in N}, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur ψ_0 . Soient $Proto = \langle convention, (u_p)_{p \in N} \rangle$ un protocole et $h \in H$ un historique. La **base d'arguments** de l'historique h (notée $Argbase(h)$) est un système d'argumentation. L'ensemble des arguments envisagés est constitué de la clôture de l'ensemble des arguments notifiés dans les tableaux d'engagements et des arguments initiaux : $Argbase(h) = Cl(Arg(CS_V(h) \cup CS_F(h)) \cup Arg(\mathcal{T}_{init}))$.

C'est à partir de cette base d'arguments que la victoire est attribuée à l'un des joueurs, c'est à dire que les gains sont déterminés. Parce que cette base est dynamique, on ne se focalise pas sur la possibilité de gagner un dialogue comme dans la théorie de la preuve dialectique. Dans une discussion critique on envisage le résultat d'un historique. Il est probable que certains arguments n'aient pas été avancés. En conséquence, cela n'a pas de sens de vouloir évaluer le résultat vis à vis de tous les arguments. On envisage uniquement les arguments qui ont été avancés. Le résultat d'une discussion critique est relatif à l'historique en cours.

Définition n° 4.3.2.XI [Correction et complétude d'un protocole]

Soient $DS_{Moves}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur ψ_0 et $Proto = \langle convention, (u_p)_{p \in N} \rangle$ un protocole. Soit $h_k = (M_1, \dots, M_k) \in H$ un historique (avec $k > 1$).

- le protocole $Proto$ est **correct** pour l'historique h_k ssi :
si $u_V(h_k) = 1$ alors $provable^{Argbase(h_k)}(\psi_0)$;
- le protocole $Proto$ est **complet** pour l'historique h_k ssi :
si $provable^{Argbase(h_k)}(\psi_0)$ et $\neg provable^{Argbase(h_{k-1})}(\psi_0)$ alors l'agent validateur peut gagner n'importe quelle extension de h_k sur la base de $Argbase(h_k)$.

On peut remarquer que la complétude d'un protocole n'exige pas que l'agent validateur gagne tous les historiques pour lesquels la formule initiale est prouvable.

Comment caractériser un protocole correct et complet ? En d'autres termes, quelles conditions doivent vérifier la convention et les gains pour qu'un protocole soit correct et complet ? Nous allons envisager une première solution dite naïve dans la section 4.3.2.3. Ce type de protocole n'est pas satisfaisant, nous envisagerons donc dans la section 4.3.2.4 une seconde solution, moins naïve.

4.3.2.3 Protocole naïf

Une première solution, dite naïve, consiste à déclarer l'agent validateur gagnant (resp. perdant) d'un historique dans la mesure où la formule initiale est (resp. n'est pas) prouvable dans la base d'arguments de l'historique.

Définition n° 4.3.2.XII [Protocole naïf]

Soit $DS_{Movers}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur ψ_0 . Le protocole $Proto = \langle convention, (u_p)_{p \in N} \rangle$ est **naïf** ssi :

- les gains u_V, u_F sont conformes à la définition 4.3.2.III ;
- les gains u_V, u_F vérifient que : $\forall h \in H (u_V(h) = 1 \Leftrightarrow \text{provable}^{\text{Argbase}(h)}(\psi_0))$.

Cette solution est illustrée dans l'exemple suivant.

Exemple n° 4.3.2.I [Protocole naïf] Le système d'argumentation AS_2 est représenté dans la figure 4.3.2.1. Les arguments sont exprimés dans une logique argumentative

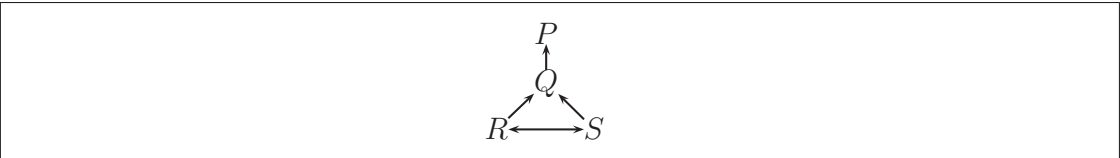


Fig. 4.3.2.I: Représentation sous forme de graphe orienté du système d'argumentation AS_2 .

de la manière suivante :

$$\begin{aligned}
 P : \text{support}(P) &= \{\phi, \phi \rightarrow \psi_0\} & \text{conclusion}(P) &= \psi_0 \\
 Q : \text{support}(Q) &= \{\pi, \pi \rightarrow \neg\phi\} & \text{conclusion}(Q) &= \neg\phi \\
 R : \text{support}(R) &= \{\beta, \beta \rightarrow \neg\pi\} & \text{conclusion}(R) &= \neg\pi \\
 S : \text{support}(S) &= \{\neg\beta, \neg\beta \rightarrow \neg\pi\} & \text{conclusion}(S) &= \neg\pi
 \end{aligned}$$

Considérons la discussion critique qui s'effectue dans le cadre du système dialectique multi-agents $DS_{Movers}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, convention, Z, (u_p)_{p \in N} \rangle$.

Considérons le protocole $Proto = \langle convention, (u_p)_{p \in N} \rangle$. Supposons que ce protocole est naïf. Considérons que la convention est définie telle que :

- les deux joueurs V et F sont autorisés à répéter les arguments de V : $M_{n+1} \in convention(h)$ si $act(M_{n+1}) = act(M_k)$ et $player(M_k) = V$;
- V n'est pas autorisé à répéter les arguments de F : $M_{n+1} \notin convention(h)$ si $act(M_{n+1}) = act(M_k)$, $player(M_k) = F$ et $player(M_{n+1}) = V$;
- F est autorisé à répéter ses propres arguments dans la mesure où ils ne sont pas dans la même ligne persuasive : $M_{n+1} \notin convention(h)$ si $player(M_{n+1}) = F$, $act(M_{n+1}) = act(M_k)$, $player(M_k) = F$ et $M_k \in line(h)$.

Considérons que les gains u_V et u_F sont définis tq si le falsificateur cite l'agent valideur alors le falsificateur gagne : si $act(M_{n+1}) = act(M_k)$, $player(M_k) = V$ et $player(M_{n+1}) = F$ alors $u_V = -1$. Ce protocole correspond à celui utilisé dans une confrontation (cf définition 3.5.0.XV). Les participants admettent conjointement cet ensemble d'arguments $Arg(\mathcal{T}_{init}) = AR_2$. Si $h_1 = (M_1)$ alors $provable^{Argbase(h)}(\Psi_0)$. Pour preuve, le système dialectique $DS_{Moves}(\psi_0, Argbase(h))$ qui porte sur ψ_0 est représenté dans la figure 4.3.2.I. Toutefois l'agent valideur n'est pas victorieux de l'historique h_1 .

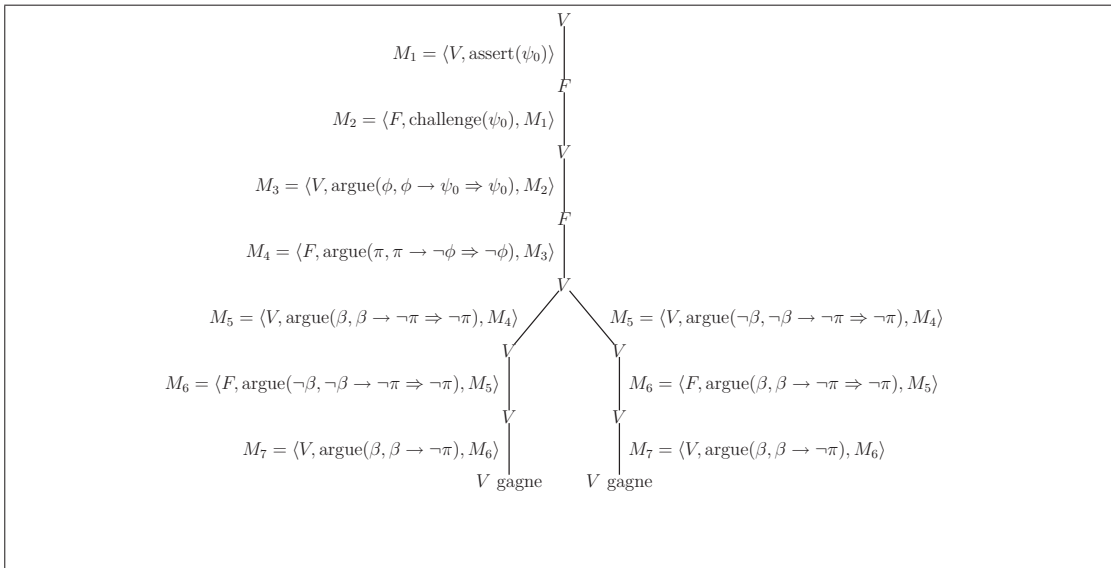


Fig. 4.3.2.II: Représentation sous forme normale extensive du système dialectique $DS_{Moves}(\psi_0, Argbase(h))$.

En effet, le falsificateur peut répondre. Toutefois la propriété de complétude nécessite que V ait une stratégie gagnante après M_1 dans la mesure où son adversaire ne peut introduire de nouveaux arguments. C'est le cas ici. Le protocole $Proto$ est correct et complet pour l'historique h_1 .

Utiliser un protocole naïf n'est pas une solution satisfaisante pour des raisons d'ordre computationnel, pragmatique et philosophique.

Primo, les gains ne sont pas facilement calculables. Cela nécessite de calculer l'historique h dans un premier système dialectique ($DS_{\text{Moves}}(\psi_0, AS)$) et puis de calculer la prouvabilité de la formule initiale ψ_0 dans un second système dialectique ($DS_{\text{Moves}}(\psi_0, \text{Argbase}(h))$). Il serait plus efficace d'évaluer à chaque nouvelle étape d'un historique la prouvabilité de la formule dans la base correspondante des arguments.

Secundo, ce calcul des gains est artificiel et purement technique. Il n'est pas suffisamment simple et transparent pour être accepté par des utilisateurs d'un système d'aide à l'argumentation.

Tertio, ce calcul de gain ne prend pas en considération la structure de la discussion en cours mais seulement la base d'arguments relative à l'historique. Il ne prend pas en considération la fonction convention. Il est, dans ces conditions, difficile de justifier de telles règles dialectiques dont l'objectif, rappelons-le, consiste à réguler le dialogue et donc à capturer une rationalité procédurale. Dans quelle mesure les règles dialectiques guident-elles efficacement un processus qui se veut rationnel si elles n'influencent pas directement le résultat ?!

4.3.2.4 Protocole bavard

Afin de caractériser un protocole complet et correct, une seconde solution, moins naïve, consiste à déclarer victorieux le joueur qui a le dernier mot dans la base d'arguments.

Définition n° 4.3.2.XIII [Protocole bavard]

Soit $DS_{\text{Moves}}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur ψ_0 . Le protocole $\text{Proto} = \langle \text{convention}, (u_p)_{p \in N} \rangle$ est **bavard** ssi :

1. la fonction *convention* est conforme à la définition 4.3.2.III. De plus, elle ne permet d'avancer que des arguments qui sont dans $\text{Argbase}(h)$;
2. les gains u_V, u_F sont conformes à la définition 4.3.2.III. Si un participant ne peut pas avancer d'arguments de $\text{Argbase}(h)$, alors la victoire est attribuée à son adversaire.

Cette solution est transparente et simple. Elle peut donc convenir à un système d'aide à l'argumentation. Malheureusement, elle ne permet de garantir ni la correction, ni la complétude des protocoles qu'ils soient à réponses uniques ou à réponses multiples. Les exemples suivants en sont la preuve.

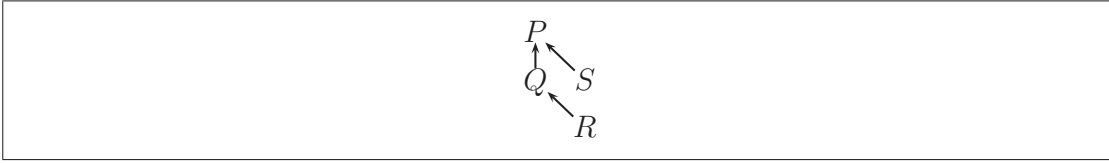


Fig. 4.3.2.III: Représentation sous forme de graphe orienté du système d'argumentation AS_5 .

Exemple n° 4.3.2.II [Protocole à réponses uniques] *Le système d'argumentation AS_5 est représenté dans la figure 4.3.2.III. Les arguments sont exprimés dans une logique argumentative de la manière suivante :*

$$\begin{aligned}
 P &: \text{support}(P) = \{\phi, \phi \rightarrow \psi_0\} \quad \text{conclusion}(P) = \psi_0 \\
 Q &: \text{support}(Q) = \{\pi, \pi \vee \xi \rightarrow \neg\phi\} \quad \text{conclusion}(Q) = \neg\phi \\
 R &: \text{support}(R) = \{\xi, \xi \rightarrow \neg\pi\} \quad \text{conclusion}(R) = \neg\pi \\
 S &: \text{support}(S) = \{\xi, \xi \vee \pi \rightarrow \neg\phi\} \quad \text{conclusion}(S) = \neg\phi
 \end{aligned}$$

Cette structure des arguments vérifie que $Cl(\{P, Q, R\}) = \{P, Q, R, S\}$ et $Cl(\{P, Q, R, S\}) = \{P, Q, R, S\}$. Considérons un protocole à réponses uniques qui soit un protocole bavard. Le dialogue orienté par un tel protocole est représenté dans la figure 4.3.2.II. On constate que V gagne l'historique $(M_1, M_2, M_3, M_4, M_5)$. Toutefois

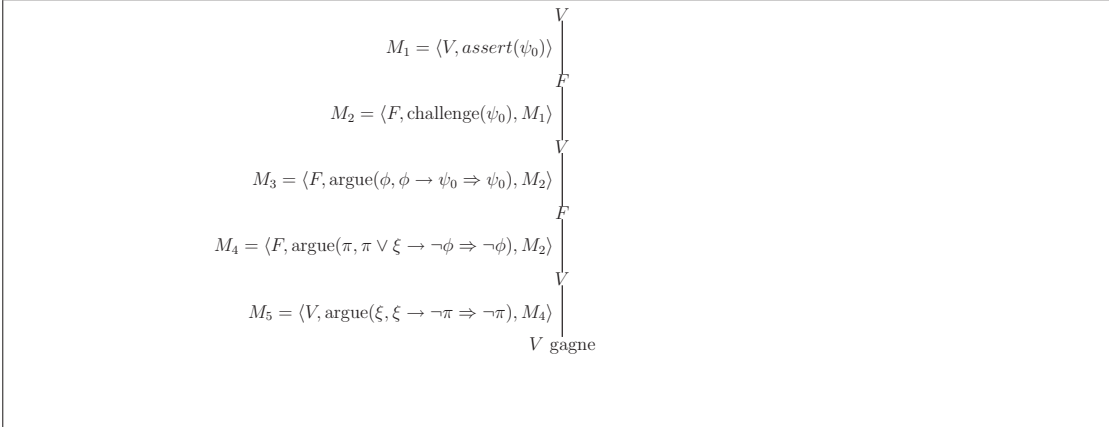


Fig. 4.3.2.IV: Représentation du dialogue orienté par un protocole bavard à réponses uniques.

P n'est pas prouvable dans la base d'arguments $\{P, Q, R, S\}$. Ce protocole n'est donc pas correct. En effet, ce protocole ne permet pas au falsificateur d'attaquer P avec S .

Cet exemple nous a permis d'illustrer la différence entre la dynamique d'une discussion critique et le caractère statique de la théorie de la preuve. Comme l'illustre l'exemple suivant, un protocole bavard à réponses multiples n'est pas nécessairement correct.

Exemple n° 4.3.2.III [Protocole à réponses multiples] Le système d'argumentation AS_6 est représenté dans la figure 4.3.2.V. Les arguments sont exprimés dans une

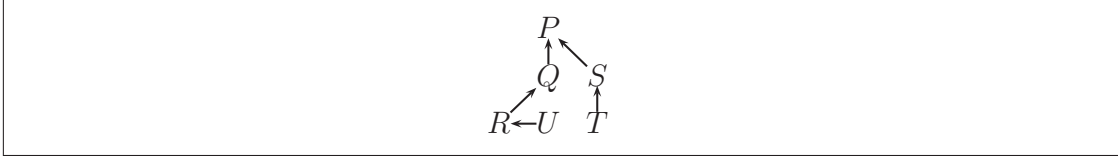


Fig. 4.3.2.V: Représentation sous forme de graphe orienté du système d'argumentation AS_6 .

logique argumentative de la manière suivante :

$$\begin{aligned}
 P &: \text{support}(P) = \{\phi, \phi \rightarrow \psi_0\} & \text{conclusion}(P) = \psi_0 \\
 Q &: \text{support}(Q) = \{\pi_1, \pi_1 \rightarrow \neg\phi\} & \text{conclusion}(Q) = \neg\phi \\
 R &: \text{support}(R) = \{\xi_1, \xi_1 \rightarrow \neg\pi_1\} & \text{conclusion}(R) = \neg\pi_1 \\
 S &: \text{support}(S) = \{\pi_2, \pi_2 \rightarrow \neg\phi \wedge \neg\xi_1\} & \text{conclusion}(S) = \neg\phi \\
 T &: \text{support}(T) = \{\xi_2, \xi_2 \rightarrow \phi\} & \text{conclusion}(T) = \phi \\
 U &: \text{support}(U) = \{\pi_2, \pi_2 \rightarrow \neg\phi \wedge \neg\xi_1\} & \text{conclusion}(U) = \neg\xi
 \end{aligned}$$

Cette structure des arguments vérifie que $Cl(\{P, Q, R, S, T\}) = \{P, Q, R, S, T, U\}$ et $Cl(\{P, Q, R, S, T, U\}) = \{P, Q, R, S, T, U\}$. Considérons un protocole à réponses multiples qui soit un protocole bavard. Le dialogue orienté par un tel protocole est représenté dans la figure 4.3.2.III. On constate que V gagne l'historique $(M_1, M_2, M_3, M_4, M_5, M_6, M_7)$.

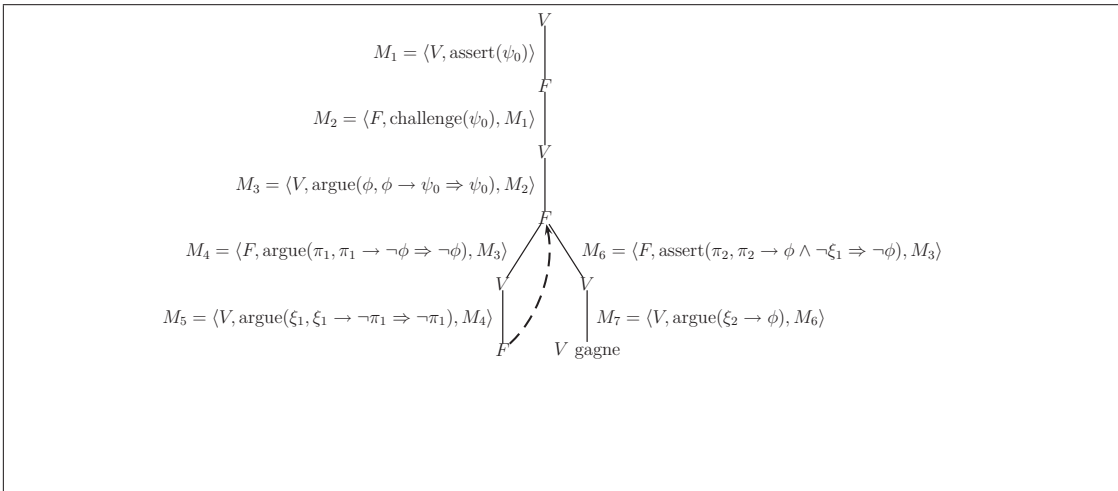


Fig. 4.3.2.VI: Représentation d'un dialogue orienté par un protocole bavard à réponses multiples.

Toutefois ψ_0 n'est pas prouvable dans la base d'arguments $\{P, Q, R, S, T\}$. Ce protocole

n'est donc pas correct. En effet, ce protocole ne permet pas au falsificateur d'attaquer R avec U.

En résumé, un protocole bavard n'est correct que dans la mesure où il permet d'avancer de nouveaux actes de langage qui attaquent de manière alternative les précédents actes de langage de l'adversaire. Le problème provient moins de la propriété de correction que celle de complétude. La propriété de complétude porte sur le futur d'un historique. Si un protocole interdit d'avancer un nouvel acte de langage pour répondre à un précédent acte de langage de l'adversaire alors ce protocole est inévitablement incomplet. En conséquence, les protocoles naïfs semblent être les seuls protocoles à réponses uniques (ou multiples) qui soient corrects et complets.

On peut envisager deux solutions pour pallier cette imperfection. Soit on définit des gains pour les protocoles à réponses uniques (ou multiples) qui garantissent la correction et la complétude de tels protocoles. Soit on propose des règles dialectiques suffisamment flexibles pour garantir la correction et la complétude d'un protocole bavard. Dans le premier cas, rien ne garantit que la définition des gains et donc l'attribution de la victoire à l'un des joueurs sera suffisamment naturelle et simple pour être intégrée dans un système d'aide à l'argumentation. C'est la raison pour laquelle la seconde solution est privilégiée.

L'objectif consiste alors à définir un protocole bavard qui soit correct et complet. Comme nous l'avons vu précédemment, les protocoles à réponses uniques et les protocoles à réponses multiples ne conviennent pas. En effet, ils ne permettent pas d'avancer n'importe quel acte de langage. Il est donc nécessaire de proposer un protocole plus permissif.

Définition n° 4.3.2.XIV [Protocole libertaire]

*Soit $DS_{Moves}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur ψ_0 . Le protocole $Proto = \langle convention, (u_p)_{p \in N} \rangle$ est **libertaire** ssi :*

1. *Proto est un protocole bavard ;*
2. *la fonction convention, conforme à la définition 4.3.2.III, vérifie que : $\forall M_k \in convention(h), act(M_k)$ est pertinent.*

Remettons à plus tard la définition technique de la pertinence d'un acte de langage. Ce type de protocole est moins restrictif qu'un protocole à réponses multiples. En effet, un joueur peut librement avancer un acte de langage qui fait référence à un des précédents coups de son adversaire. Le coup auquel il fait référence n'appartient pas nécessairement à la ligne persuasive en cours. Comme nous l'avons fait remarquer

précédemment, utiliser ce type de protocole entraîne une surcharge computationnelle. C'est le prix à payer pour être complet.

Comment définir la pertinence d'un acte de langage ? On doit au préalable étiqueter les actes de langage dans un arbre de dialogue. On est alors en mesure de distinguer les coups qui sont en discussion et ceux qui ne le sont pas. Le statut d'un coup est assigné sur la base des coups émis dans un dialogue sans prendre en considération la base des arguments.

Définition n° 4.3.2.XV [Statut discursif d'un coup]

Soit $DS_{Moves}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur ψ_0 . Soit $h_k \in H$ un historique, M_i est **en discussion** dans l'historique h_k (avec $i \leq k$) ssi tous les coups qui y répondent n'ont pas été émis dans ce dialogue. Dans le cas contraire, ce coup n'est pas en discussion. On note $discussed_{h_k}(M_i)$ le statut discursif du coup $M_i \in h_k$ dans l'historique h_k .

On dit alors d'un coup qu'il est pertinent ssi il modifie le statut discursif du coup initial.

Définition n° 4.3.2.XVI [Pertinence d'un coup]

Soit $DS_{Moves}(\psi_0, AS) = \langle N, H, T, (CS_p)_{p \in N}, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur ψ_0 . Soit $h_{k-1} = (M_1, \dots, M_{k-1}) \in H$ un historique, M_k est **pertinent** dans h_{k-1} ssi $discussed_{h_{k-1}}(M_1) \neq discussed_{h_k}(M_1)$.

Henry Prakken a démontré dans [50] que le protocole libertaire est correct et complet. Pour des raisons de concision nous ne détaillerons pas cette longue démonstration ici. Toutefois, l'exemple permet d'illustrer la complétude et la correction d'un protocole libertaire.

Exemple n° 4.3.2.IV [Protocole libertaire] Si on reprend l'exemple 4.3.2.III en utilisant un protocole libre, alors on constate que le dialogue orienté par un tel protocole et représenté dans la figure 4.3.2.IV est gagné par F . ψ_0 n'est pas prouvable dans la base d'arguments $\{P, Q, R, S, T\}$. Ce protocole est correct et complet. En effet, ce protocole permet à l'agent falsificateur d'attaquer R avec U .

En résumé, un système dialectique multi-agents est un dispositif au travers duquel deux agents s'affrontent à propos de la validité d'une formule. Contrairement à un jeu argumentatif d'évaluation sémantique, les participants d'une discussion critique sont des agents munis de leurs propres arguments. Dans cette perspective, la base d'arguments à partir de laquelle un argumentateur est déclaré victorieux n'est pas fixée *a priori*.

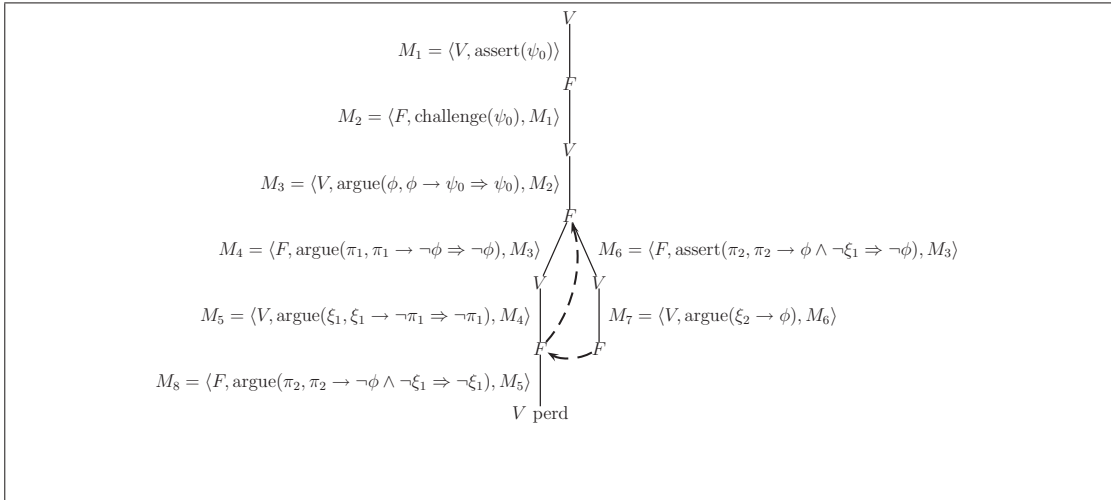


Fig. 4.3.2.VII: Représentation du dialogue orienté par un protocole libre.

4.4 Modèle de dialogue

Dans ce qui précède, nous avons vu qu'en s'inspirant de la dialectique, les modèles d'interaction directe entre agents ont intégré la notion d'engagement et celle d'argument. Nous présentons ici un travail qui permet de faire le lien entre les arguments qui sont échangés dans un dialogue et les engagements qui sont pris dans un dialogue.

Dans [10], Jamal Bentahar, Bernard Moulin et Brahim Chaib-draa proposent un cadre formel qui permet de représenter les actions des agents au cours des dialogues. Ces actions sont interprétées en termes de manipulation et de positionnement par rapport à des engagements sociaux. D'une part, les agents doivent être capables de justifier les faits sur lesquels ils s'engagent. D'autre part, les agents doivent être capables de justifier leurs actions sur les engagements.

Nous allons dans un premier temps présenter le modèle de dialogue proposé (cf section 4.4.1). Dans un second temps, nous aborderons rapidement les relations entre les arguments et les engagements (cf section 4.4.2).

4.4.1 Le modèle de dialogue

[10] propose un modèle de dialogue qui combine trois approches de l'interaction : l'approche mentaliste, l'approche argumentative et l'approche sociale.

L'approche mentaliste (cf section 1.5), qui considère une énonciation comme une action intentionnelle, définit la pragmatique des messages en termes d'états mentaux. L'approche argumentative modélise l'interaction par une joute verbale entre des participants qui tentent de persuader leur partenaire en échangeant des argu-

ments (cf section 4.3.2). L'approche sociale considère le dialogue au travers d'une structure bilatérale qui engage deux participants, un initiateur et un partenaire, dans un échange de messages ordonnés et cohérents orientés par des règles pour établir un engagement social (cf section 4.2.2).

Le modèle de dialogue proposé dans [10] est représenté dans la figure 4.4.1.I. Ce modèle, représenté à droite de la figure, est stratifié en trois couches. Chacune de ces couches correspond à un niveau d'abstraction différent. La couche de bas niveau, intitulée couche de dialogue, est constituée des actes de langage qui sont échangés par les agents au cours des dialogues. La couche supérieure permet de gérer les engagements et les arguments relatifs aux dialogues. Alors que les actes de langage sont directement observables, les engagements et les arguments sont déduits des actes échangés. La troisième et dernière couche permet de prendre en considération les états mentaux des agents ainsi que les relations sociales qui les lient les uns aux autres. Cette couche permet de gérer les aspects privés aux agents.

Ce modèle de dialogue est mis en relation avec une architecture d'agent dialogique représentée à droite de la figure 4.4.1.I. Cette architecture est décomposée en trois modèles. Le modèle mental manipule les croyances, les buts et les intentions des agents (*Belief, Desire, Intention*). Le modèle social permet de manipuler les conventions et les engagements. Alors qu'un engagement est public, un état mental est propre à l'agent. Toutefois ces notions ne sont pas indépendantes. Un engagement social révèle l'état mental d'un agent. C'est la raison pour laquelle un agent est doté de capacité de raisonnement. À partir de ses états mentaux et de son modèle social, le raisonnement de l'agent va guider son comportement. Il va argumenter et s'engager socialement. De plus, un agent est muni de connaissances générales qui lui permettent de constituer un terrain commun avec ses partenaires dans les dialogues.

4.4.2 Les relations entre engagements et arguments

Dans [10], Jamal Bentahar, Bernard Moulin et Brahim Chaib-draa explicitent les relations qu'il existe entre les arguments et les engagements.

Dans ce travail, les auteurs étendent la définition d'engagement social (cf définition 4.2.2.I) de la manière suivante :

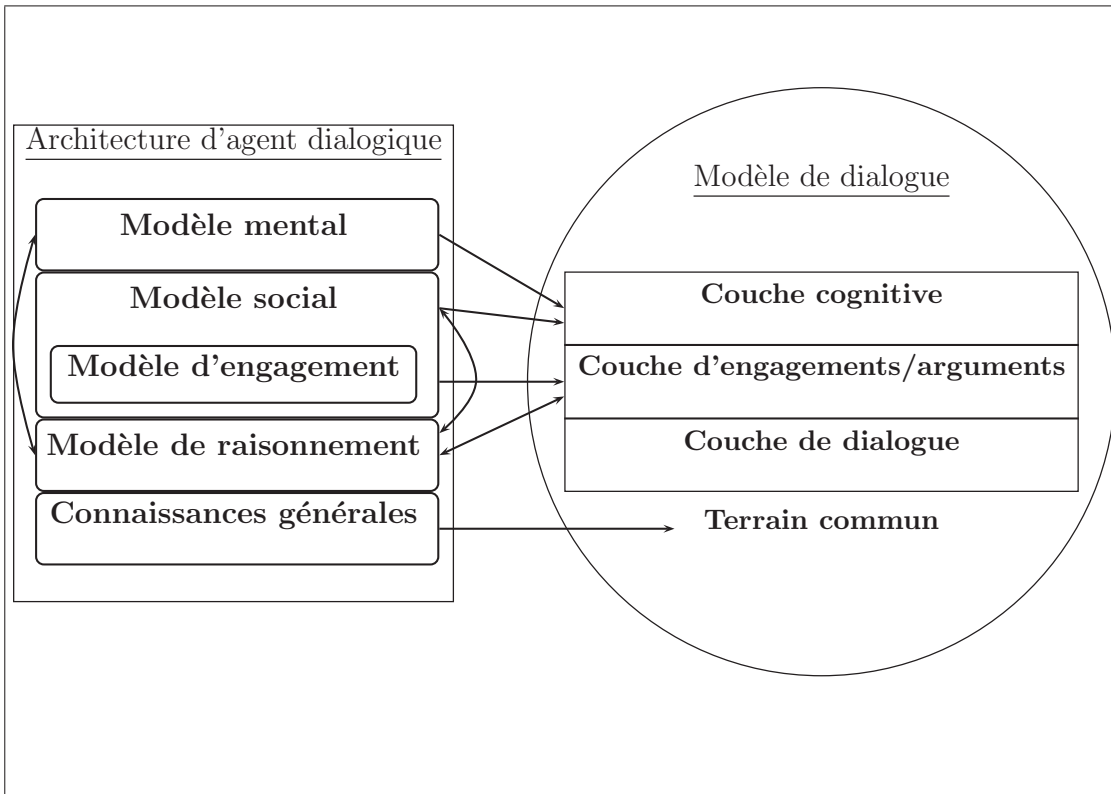


Fig. 4.4.1.I: Liens entre architecture d'un agent dialogique et modèle de dialogue

Définition n° 4.4.2.I [Engagement social]

On appelle **engagement social** le prédicat suivant :

$$C(x, Y, t_C, \alpha, t_\alpha, s_x, s_y) \quad (4.4.2.I)$$

D'après ce prédicat, l'agent x , appelé **débiteur**, s'engage au temps t_C vis à vis des agents Y , appelés **crédeurs**, à réaliser l'action α , appelée **contenu**, au temps t_α . Si l'agent x décide d'annuler l'engagement, il se verra infliger la **sanction** s_x alors que les agents Y qui libèrent l'engagement se verront infliger la **sanction** s_y .

D'une part, un crédeur ne s'engage pas vis à vis d'un agent mais vis à vis d'un ensemble d'agents, appelés crédeurs. D'autre part, ce travail distingue la dynamique de l'engagement et la dynamique du contenu de l'engagement. En effet, cette définition distingue t_α la date butoire à laquelle l'engagement doit être rempli et la date t_C associée à l'engagement. C'est un instant ou un intervalle de temps ($[t_C^{inf}, t_C^{sup}]$) au cours duquel l'engagement est en vigueur.

L'engagement est muni d'un cycle de vie quasi identique à celui décrit dans la figure

4.2.2.I. Dans ce travail, un engagement ne peut pas être libéré. L'état **Released** n'existe pas.

Le contenu d'un engagement social est également dynamique. Lorsqu'un engagement est actif, son contenu est **soumis** (noté **Submitted**). Le contenu d'un engagement peut alors être **modifié** (noté **Changed**), **mis en doute** (noté **Challenged**), **justifié** (noté **Argued**), **accepté** (noté **Accepted**) ou **refusé** (noté **Refused**). La figure 4.4.2.I représente le cycle de vie du contenu d'un engagement sous la forme d'un automate fini (cf définition 1.4.2.I). L'état **Submitted** est initial. Les états **Accepted** et **Refused** sont finaux.

Afin de modéliser le dynamisme des dialogues, les actes de langage sont interprétés en terme d'action sur les engagements ou sur leur contenu. On distingue d'une part les actions qui peuvent être réalisées par le créateur sur l'engagement et d'autre part, les actions qui peuvent être réalisées par le créateur ou les débiteurs sur le contenu. Seuls les actes de langage émis par le débiteur agissent sur l'engagement. Ces actes permettent de créer, d'annuler, de remplir ou de violer l'engagement. À l'inverse, les créateurs et le débiteur peuvent agir sur le contenu d'un engagement. Ils se positionnent par rapport au contenu. Le débiteur peut faire des concessions sur le contenu de l'engagement. Il peut également le justifier si le créateur le met en doute. Le créateur peut mettre en doute, accepter ou refuser le contenu d'un engagement. Les locutions utilisées ont une sémantique argumentative (cf section 4.3.2).

De plus, [10] propose un formalisme, que nous ne détaillerons pas ici, qui permet d'analyser les dialogues et qui peut aider les agents dialogiques à prendre part à des dialogues consistants.

En résumé, le cadre formel proposé ici permet de lier les arguments échangés et les engagement pris dans les dialogues.

4.5 Synthèse

Ce chapitre a été consacré aux **modèles de dialogue entre agents**, *i.e.* les modèles d'interaction directe entre agents qui s'inspirent de la dialectique.

Un **système dialectique multi-agents** est un dispositif à travers lequel deux **agents argumentatifs**, chacun muni de son propre argumentaire, s'échangent des **coups dialogiques**. Un agent argumentatif affirme une hypothèse dans la mesure où il dispose d'un argument en sa faveur. En d'autres termes, les actes de langage ont une sémantique argumentative. À l'inverse, les tableaux d'engagement répertorient l'ensemble des hypothèses sur lesquelles les agents s'engagent. Initialement vides, les

tableaux d'engagement conservent la trace des engagements au cours des échanges. En d'autres termes, les actes de langage ont une sémantique sociale. Les coups dialogiques échangés doivent être conformes à une convention, *i.e.* un ensemble de **règles dialectiques**. En d'autres termes, ces règles capturent l'aspect conventionnel de la pragmatique du langage de communication.

Pour les trois raisons suivantes, cette approche permet de modéliser le dialogue au sens politique, c'est à dire une prise de décision concertée.

1. La sémantique des actes de langage est argumentative. Le raisonnement des agents s'appuie sur une logique argumentative. Ce modèle de raisonnement permet aux agents de gérer les incohérences entre leurs propres croyances et les informations transmises par d'autres agents.
2. La sémantique des actes de langage est sociale. Les engagements pris sont publics et objectifs. Un agent qui assiste au dialogue confère le même sens aux actes de langage et réalise les mêmes inférences, qu'il en soit le destinataire ou non.
3. Les règles dialectiques constituent une alternative au protocole d'interaction. D'une part, ce sont des structures collaboratives de communication flexible. Elles permettent de coordonner l'activité dialogique des agents. D'autre part, parce que le cadre est bien défini, on peut caractériser le résultat obtenu au terme du dialogue.

Dans le chapitre suivant, nous allons évaluer dans quelle mesure les travaux évoqués ici permettent ou non de répondre à notre problématique.

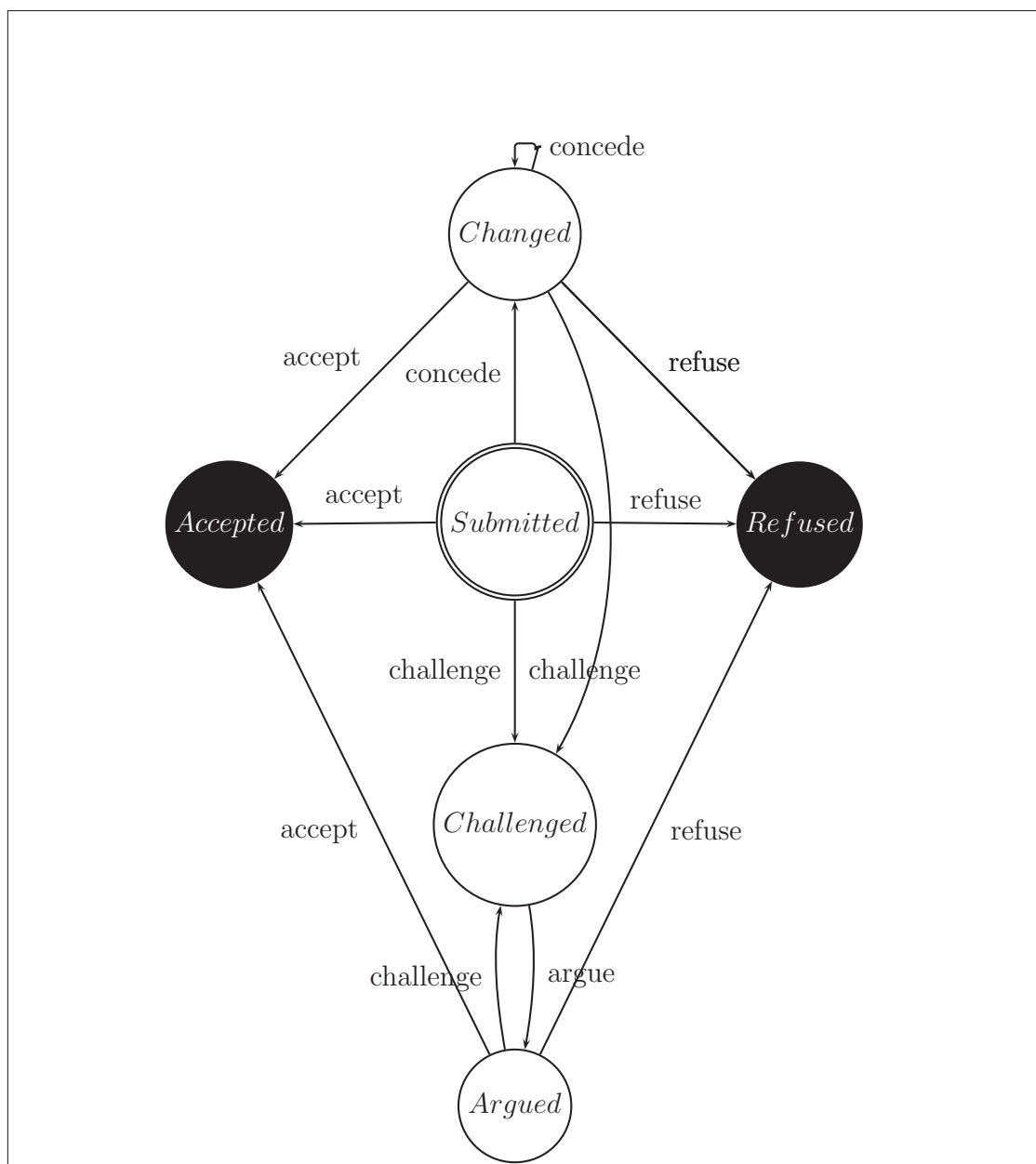


Fig. 4.4.2.I: Cycle de vie du contenu d'un engagement

“In computer science, important concepts usually come with a plethora of alternative characterizations”.

Christos Papdimitriou

Chapitre 5

Conclusion de l'état de l'art

Sommaire

5.1	Modèles de raisonnement argumentatif	115
5.2	Modèles de dialogue entre agents	117

Dans cette partie, nous avons dégagé les caractéristiques essentielles que doit posséder un modèle de dialogue entre agents pour formaliser une décision collective et débattue. Dans cette perspective, cette partie met en exergue les limites de l'approche mentalistique développée dans les modèles d'interaction directe entre agents et précise comment l'approche dialectique y répond.

Nous allons résumer ce en quoi les modèles de raisonnement argumentatif (cf section 5.1) et les modèles de dialogue (cf section 5.1) évoqués dans cette partie répondent (ou non) à nos exigences.

5.1 Modèles de raisonnement argumentatif

L'approche mentalistique ne propose pas de modèle de raisonnement qui permet à un agent de gérer les incohérences entre ses propres croyances et les informations reçues. Dans le chapitre 2, nous avons montré que l'argumentation constitue un modèle de raisonnement adapté au processus cognitif d'un agent autonome et social pour gérer les interactions entre les arguments internes qui explicitent ses croyances et les arguments externes qui créditent des croyances contradictoires provenant d'autres agents.

Les modèles formels de raisonnement argumentatif abordés dans le chapitre 2 sont représentés dans la figure 5.1.0.I.

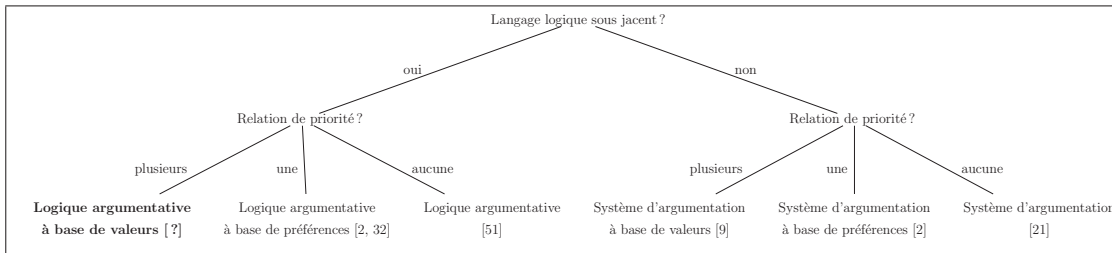


Fig. 5.1.0.I: Arbre de décision pour classer les modèles de raisonnement argumentatif

1. Nous avons parcouru dans la section 2.3 les travaux qui considèrent les arguments comme des entités abstraites. Ces travaux sont représentés dans le sous-arbre droit.
 - Le système d’argumentation proposé par Phan Minh. Dung [21], présenté dans la section 2.3.1, est un modèle formel de raisonnement au travers duquel des arguments liés les uns aux autres peuvent être catégorisés. Il s’appuie sur les trois notions suivantes : le concept d’argument, une relation de contradiction et l’acceptabilité des arguments. Ce système est représenté dans la feuille droite du sous-arbre droit.
 - Le système d’argumentation à base de préférences proposé par Leila Amgoud et Claudette Cayrol [2, 1], présenté dans la section 2.3.2, est une extension du système d’argumentation de Phan Minh. Dung [21]. L’adjonction d’une relation de priorité a permis de doter les arguments d’une force. Ce système est représenté dans la feuille centrale du sous-arbre droit.
 - Le système d’argumentation à base de valeurs proposé par Trévor Bench-Capon [9] et présenté dans la section 2.3.3 est également une extension du système d’argumentation de Phan Minh. Dung [21]. L’adjonction de plusieurs relations de priorité permet de doter les arguments d’une force qui dépend de l’audience concernée. Ce système est représenté dans la feuille gauche du sous-arbre droit.
2. Nous avons parcouru dans la section 2.4 les travaux qui considèrent les arguments non plus comme des entités abstraites mais comme un schéma logique qui repose sur une base de connaissances. On distingue alors dans un argument la prémisse de la conclusion. Un argument est en faveur d’une conclusion si la conclusion est une conséquence logique de la prémisse. Ces travaux sont représentés dans le sous-arbre gauche.
 - La logique argumentative proposée par Michael Schroeder [51], abordée dans la section 2.4.1, s’appuie sur la structure des arguments pour introduire différentes relations de contradiction. Cette logique est représentée dans la feuille droite du sous-arbre gauche.

- Que cela soit dans la logique argumentative proposée par Leila Amgoud et Claudette Cayrol [1, 2] abordée dans la section 2.4.2 ou dans celle proposée par Antonis Kakas et Pavlos Moraïtis [32, 42] abordée dans la section 2.4.3, les arguments sont munis d’une force et d’une structure. Ces logiques sont représentées dans la feuille centrale du sous-arbre gauche.

Considérant les connaissances comme des représentations de l’état du monde, nous souhaitons les modéliser à l’aide de formules d’un langage de logique classique. Ainsi, il nous faut disposer d’une logique argumentative. Considérant que les agents qui disposent des mêmes connaissances peuvent avoir des points de vue divergents, nous devons considérer simultanément plusieurs relations de priorité. **Ainsi nous souhaitons disposer d’une logique argumentative à base de valeurs.** À notre connaissance, aucun travail ne répond à cette double exigence. Cette vacuité est représentée dans la feuille gauche du sous-arbre gauche.

5.2 Modèles de dialogue entre agents

L’approche mentalistique confère aux actes de langage une sémantique privée et subjective. De plus, les protocoles d’interaction sont rigides et leur seule spécification ne permet pas de garantir l’obtention d’un résultat. Nous avons montré dans le chapitre 4 que l’approche dialectique permet de dépasser ces limites.

Les modèles de dialogue abordés dans le chapitre 4, c’est-à-dire les modèles d’interaction directe entre agents qui s’inspirent de la dialectique, sont représentés dans la figure 5.2.0.II.

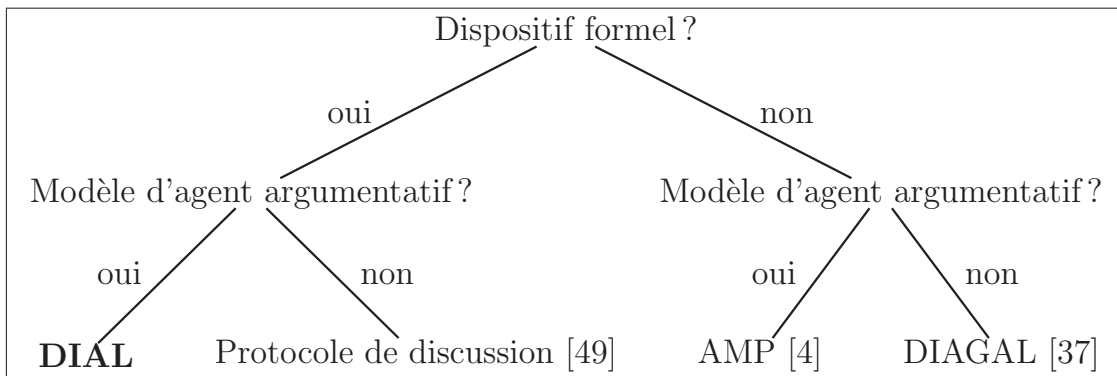


Fig. 5.2.0.II: Arbre de décision pour classer les modèles de dialogue entre agents.

- Le langage de communication d’agents DIAGAL [47] présenté dans la section 4.2.2 s’inspire de la dialectique. En effet, il s’appuie sur la notion d’engagement social qui permet notamment de formaliser les règles dialectiques. Puisqu’il ne propose

pas de modèle d'agents, DIAGAL est représenté dans la feuille droite du sous-arbre droit.

- Le modèle AMP [4] présenté dans la section 4.3.1 s'inspire également de la dialectique. Contrairement au précédent, ce travail s'appuie sur un modèle d'agent qui spécifie dans quelle mesure les actes peuvent être énoncés et comment ils sont interprétés. C'est la raison pour laquelle, ce modèle est représenté dans la feuille gauche du sous-arbre droit.
- Le modèle proposé par Henry Prakken [49] présenté dans la section 4.3.2 circonscrit un cadre au travers duquel des agents dialoguent. Un tel dispositif formel permet de garantir l'obtention d'un résultat (*i.e.* la correction) et de préjuger de sa qualité (*i.e.* la complétude). Contrairement au travail précédemment évoqué aucun modèle d'agent n'est proposé. C'est pour ces deux raisons que ce travail est représenté dans la feuille droite du sous-arbre gauche.

Souhaitant fournir un modèle de dialogue complètement spécifié, nous devons décrire l'ensemble des règles qui permettent aux agents de gérer les dialogues. **Ainsi nous souhaitons disposer d'un modèle d'agent argumentatif.** Souhaitant garantir l'obtention d'un résultat et évaluer sa qualité, nous devons proposer un cadre formel dans lequel les dialogues sont joués. **Ainsi nous souhaitons disposer d'un système dialectique multi-agents.** À notre connaissance, aucun modèle de dialogue ne répond à cette double exigence. Cette vacuité est représentée dans la feuille gauche du sous arbre droit.

<p>En résumé, le modèle DIAL présenté dans la partie suivante doit, pour répondre à notre objectif, circonscire un cadre formel dans lequel les agents dialoguent. Ce cadre est appelé système dialectique multi-agents. Le modèle d'agent argumentatif devra spécifier dans quelle mesure les actes peuvent être énoncés et comment ils sont interprétés. Le modèle formel de raisonnement argumentatif doit être une logique argumentative à base de valeurs.</p>
--

Deuxième partie

Modèle DIAL

« The partisan, when he is engaged in a dispute, cares nothing about the rights of the question, but is anxious only to convince his hearers of his own assertions. »

Plato, Dialogues, Phaedo

Chapitre 6

Introduction du modèle

Suite aux exigences identifiées dans la partie précédente, nous proposons ici un modèle de dialogue entre agents qui y répond. Ce modèle, intitulé **DIAL** (*DIALogue Is an Argumentative Labour*¹), contribue à la formalisation d'une prise de décision concertée.

Afin que les agents puissent gérer les interactions entre des arguments éventuellement contradictoires provenant de différentes sources, ils seront munis d'un modèle formel de raisonnement argumentatif. Pour pouvoir exprimer des connaissances représentant l'état du monde, ce modèle de raisonnement s'appuie sur un langage logique sous-jacent. Dans le dessein de munir chaque agent de sa propre échelle de valeur, chacun d'eux doit évaluer indépendamment la force des arguments. À cette intention, nous définissons dans le chapitre 7 la logique argumentative qui constitue le modèle de raisonnement des agents. Cette **logique argumentative à base de valeurs** permet de gérer les interactions entre des arguments contradictoires qui sont plus ou moins forts selon l'agent envisagé.

Ayant défini le modèle de raisonnement des agents, nous devons, afin de proposer un modèle de dialogue complet, spécifier les éléments du modèle DIAL qui précisent comment les actes peuvent être énoncés et comment ils sont interprétés. À cette intention, nous présentons, dans le chapitre 8, **un modèle d'agent argumentatif** qui leur permettent d'échanger leurs connaissances et de raisonner conjointement.

Ayant défini les éléments du modèle DIAL qui permettent aux agents d'échanger des informations et de raisonner conjointement, nous devons, afin de garantir l'obtention d'un résultat au terme du dialogue et d'évaluer sa qualité, spécifier les élé-

¹Labeur = Travail pénible demandant un effort soutenu et de longue haleine, une grande ténacité. Au sens figuré, un processus continu ou progressif, de transformation, de maturation. On remarque que labeur est employé comme synonyme de travail dans le style soutenu ou dans les contextes accordant une valeur morale au travail.

ments du modèle DIAL qui permettent de gérer le flux d'information en régulant l'enchaînement des messages afin de mener à bien l'interaction. À cette intention, nous proposons dans le chapitre 9 un cadre formel intitulé **système dialectique multi-agents** dans lequel les dialogues sont joués.

« Convaincre quelqu'un c'est le vaincre par des arguments forts. »

Anonyme

Chapitre 7

Système d'argumentation multi-agents

Sommaire

7.1	Introduction	123
7.2	Langage sous-jacent	124
7.3	Arguments	127
7.4	Relations entre arguments	129
7.4.1	Relation d'attaque	129
7.4.2	Relations de défaite	130
7.5	Acceptabilité des arguments	132
7.6	Synthèse	134

7.1 Introduction

Nous proposons dans ce chapitre une logique argumentative qui constitue le modèle de raisonnement des agents. Cette logique argumentative permet de gérer les interactions entre des arguments contradictoires qui sont plus ou moins forts selon l'agent envisagé.

Afin de présenter cette logique argumentative, nous allons aborder tour à tour chacune des cinq notions de base mises en exergue dans la section 2.5. Nous allons dans un premier temps définir le langage logique sous-jacent (cf section 7.2). Ce langage permet d'exprimer des connaissances à propos desquelles les agents ont des inclinaisons. Dans un second temps, nous porterons notre attention sur la structure interne des arguments (cf section 7.3). Nous présenterons dans la section 7.4 les relations

qui lient les arguments. Ces relations entre arguments permettent d'aboutir à une catégorisation des arguments selon différentes classes d'acceptabilité (cf section 7.5).

7.2 Langage sous-jacent

Le langage sous-jacent à notre logique argumentative est un langage logique. Les connaissances ainsi exprimées sont rassemblées dans une théorie. D'une part, une théorie compile l'ensemble des connaissances des agents. D'autre part, chacun des agents est associé à une relation de priorité qui explicite ses propres inclinaisons.

Les connaissances sont des représentations de l'état du monde qui peuvent être vraies ou fausses. C'est la raison pour laquelle nous les modélisons à l'aide de formules d'un langage de logique classique (cf Annexe A). Ce langage est : soit un langage propositionnel ; soit un langage du premier ordre. \mathcal{L} dénote un tel langage. Les agents doivent pouvoir raisonner à partir de ces connaissances. À cet effet, ils utilisent un système de preuves (cf Annexe A.4). \vdash dénote la déduction classique.

Ces connaissances doivent être rassemblées. Nous utilisons à cet effet, comme dans les travaux d'Antonis Kakas et de Pavlos Moraitis [32, 42] (cf section 2.4.3), un programme logique.

Définition n° 7.2.0.II [Programme logique]

Un **programme de logique** est un ensemble de clauses de programme du type :

$$r : L_0 \leftarrow L_1, \dots, L_n \quad (7.2.0.I)$$

avec L_i un littéral ($\forall 0 \leq i \leq n$).

L_0 est appelé la **tête** de la clause de programme r . On le note $L_0 = \text{head}(r)$.

L'ensemble des autres littéraux constitue le **corps** de la clause. On le note : $\text{body}(r) = \{L_1, \dots, L_n\}$.

Un littéral est un atome A ou la négation d'un atome $\neg A$. On a explicitement $L = \neg\neg L$. Leila Amgoud [1] utilise la notion de base de connaissances (cf section 2.4.2). Nous nous sommes restreints ici aux clauses de Horn. La formulation d'un programme logique est proche du langage Prolog¹ et donc de l'implémentation qui pourra en être proposée.

Les connaissances peuvent représenter des faits. On utilise alors des clauses unitées, i.e. des clauses sans corps (par exemple, $r_1 : \text{homme}(\text{Socrate}) \leftarrow$). Les connaissances

¹La plus connue des implémentations standards et libres de Prolog est SWI-Prolog <http://www.swi-prolog.org/>

peuvent représenter des règles. On utilise alors des clauses de programme avec un corps (par exemple, $r_2(x) : \text{mortel}(x) \leftarrow \text{homme}(x)$). Nous appellerons **théorie**, un ensemble fini de clauses. Ces clauses, qu'elles possèdent ou non un corps, seront appelées des **règles**. Pour des raisons computationnelles évidentes, nous considérons ici que l'ensemble des variables du langage ont des domaines de définition finis.

Comme nous le définissons ci-dessous, une théorie argumentative multi-agents est constituée d'un ensemble de règles et d'un ensemble de valeurs à partir desquelles un ensemble d'agents argumente.

Définition n° 7.2.0.III [Théorie argumentative multi-agents]

Soit $\mathcal{U}_A = \{ag_1, \dots, ag_n\}$ un ensemble d'agents. La **théorie argumentative multi-agents** est un triplet $\mathcal{AT}_{\mathcal{U}_A} = \langle \mathcal{T}, V, \text{promote} \rangle$ où :

- \mathcal{T} est une **théorie** c'est à dire un ensemble de clauses de programme appelées **règles** ;
- V est un ensemble fini de **valeurs** $\{v_1, \dots, v_t\}$;
- $\text{promote} : \mathcal{T} \rightarrow V$ est une fonction qui met en relation les règles et les valeurs.

Pour tous $r \in \mathcal{T}$, $\text{promote}(r) = v$.

On dit qu'une règle r prône une valeur v ssi $\text{promote}(r) = v$.

Les inclinaisons reflètent le ressentiment des agents. Selon Perelman [15], les ressentiments des agents diffèrent selon l'échelle de valeurs à laquelle ils se réfèrent. C'est la raison pour laquelle nous considérons, comme dans les travaux de Bench-Capon [9] (cf section 2.3.3), un ensemble de valeurs. Ces valeurs permettent de stratifier la théorie en un ensemble fini de couches. Chacune d'elles prône une valeur.

Dans la suite de ce manuscrit $\mathcal{U}_A = \{ag_1, \dots, ag_n\}$ dénote l'ensemble des agents du système. Ils se différencient les uns des autres par l'importance qu'ils affectent aux valeurs. En conséquence, on associe à chacun des agents une relation de priorité propre à l'agent.

Définition n° 7.2.0.IV [Relation de priorité]

Soient $\mathcal{AT}_{\mathcal{U}_A} = \langle \mathcal{T}, V, \text{promote} \rangle$ une théorie argumentative multi-agents et $ag_i \in \mathcal{U}_A$ l'un des agents du système. La **relation de priorité** associée à l'agent ag_i (notée \ll_i) est une relation d'ordre strict et total sur V .

Cette relation est asymétrique et transitive (cf annexe C). Contrairement aux travaux d'Antonis Kakas et de Pavlos Moraitis, les inclinaisons d'un agent ne sont pas exprimées à l'aide de règles mais d'une simple relation binaire sur les valeurs (cf définition 2.4.3.III). Les inclinaisons des agents contrairement à leurs connaissances n'ont pas vocation à être partagées ou dynamiques.

Plus formellement, on associe à chacun des agents une théorie argumentative. Elle est constituée d'une part d'une théorie argumentative multi-agents et d'autre part de la relation de priorité qui lui est propre.

Définition n° 7.2.0.V [Théorie argumentative mono-agent]

Soit $ag_i \in \mathcal{U}_A$, un agent. La **théorie argumentative de l'agent** ag_i est un quadruplet $\mathcal{AT}_i = \langle \mathcal{T}, V, \text{promote}, \ll_i \rangle$ où :

- $\mathcal{AT}_{\mathcal{U}_A} = \langle \mathcal{T}, V, \text{promote} \rangle$ est une théorie argumentative multi-agents telle que nous l'avons définie précédemment ;
- \ll_i est la relation de priorité associée à l'agent ag_i telle que nous l'avons définie précédemment.

L'échelle de valeurs à laquelle un agent se réfère permet de valuer les règles argumentatives. Ainsi, les règles qui prônent la même valeur sont équi-prioritaires et prioritaires sur celles qui prônent une valeur plus faible.

On est alors en mesure d'évaluer la priorité d'un ensemble de règles. On appelle niveau de priorité d'un sous-ensemble non vide de règles, la valeur la moins prioritaire prônée par l'une des règles.

Définition n° 7.2.0.VI [Niveau de priorité]

Soit $\mathcal{AT}_i = \langle \mathcal{T}, V, \text{promote}, \ll_i \rangle$ la théorie argumentative associée à l'agent $ag_i \in \mathcal{U}_A$. Soit $T \subseteq \mathcal{T}$ un ensemble non-vide de règles. Le **niveau de priorité de T** (noté $\text{level}(T)$) est défini tq :

$$\text{level}(T) = v_j \in V \Leftrightarrow \exists r \in T \forall r' \in T - \{r\} (\text{promote}(r) = v_j \ll_i \text{promote}(r'))$$

L'exemple présenté ci-dessous permet d'illustrer l'ensemble de ces définitions.

Exemple n° 7.2.0.I [Théorie argumentative multi-agents] Deux adhérents d'une association loi 1901 doivent, au cours d'une assemblée générale, se prononcer pour désigner le futur président de cette association.

La théorie argumentative de l'agent ag_1 (respectivement de l'agent ag_2) est représentée dans la partie gauche (respectivement dans la partie droite) de la figure 7.2.0.I. Ces deux agents partagent une théorie, i.e. un ensemble de règles (r_{11}, \dots, r_6) et un ensemble de valeurs (v_1, \dots, v_6) .

Ils doivent choisir parmi deux candidats : jack et jose. Les règles r_{11} et r_{21} relatives à l'objectif prônent la valeur v_1 . Les règles $r_{12}(x)$ et $r_{22}(x)$ qui font appel au sens commun,

\ll_1	V	\mathcal{T}	\ll_2	V	\mathcal{T}
↑	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	↑	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$
	v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$		v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$
	v_6	$r_6 : \text{president_actuel}(\text{jack}) \leftarrow$		v_3	$r_3(x) : \text{president}(x) \leftarrow \text{president_actuel}(x)$
	v_5	$r_5 : \text{prison}(\text{jose}) \leftarrow$		v_4	$r_4 : \text{escroc}(\text{jack}) \leftarrow$
	v_4	$r_4 : \text{escroc}(\text{jack}) \leftarrow$		v_5	$r_5 : \text{prison}(\text{jose}) \leftarrow$
	v_3	$r_3(x) : \text{president}(x) \leftarrow \text{president_actuel}(x)$		v_6	$r_6 : \text{president_actuel}(\text{jack}) \leftarrow$

Fig. 7.2.0.I: Théories argumentatives mono-agent

prônent la valeur v_2 . Les quatre règles suivantes (r_3 , r_4 , r_5 et r_6) sont des opinions, *i.e.* des connaissances indépendantes les unes des autres. Chacune d'elle fait référence à une valeur qui lui est propre (respectivement v_3 , v_4 , v_5 et v_6).

Les deux agents se réfèrent à des échelles de valeurs différentes. Selon un agent, une valeur située au-dessus d'une autre valeur est considérée comme davantage prioritaire. Ces deux agents sont d'accord pour considérer l'objectif comme prioritaire et les règles de sens commun comme secondaires mais prioritaires par rapport aux autres connaissances. Leur point de vue sur les compétences et les qualités des candidats divergent par les priorités qu'ils affectent aux valeurs v_3 , v_4 , v_5 et v_6 .

En résumé, le langage sous-jacent à notre logique argumentative est un langage logique. Contrairement aux inclinaisons propres aux agents, les connaissances communes sont des expressions bien formées de ce langage logique. Nous allons utiliser ce langage pour définir les arguments.

7.3 Arguments

Un argument est construit à partir d'une théorie. Il est constitué d'une conclusion et d'une prémisse, *i.e.* un ensemble de règles à partir desquelles la conclusion est déduite. Dans une théorie argumentative multi-agents, l'argumentaire est construit à partir des règles partagées par les agents.

Nous formalisons le concept d'argument de la manière suivante :

Définition n° 7.3.0.VII [Argument]

Soit T une théorie. Un **argument** est un couple $A = \langle P, c \rangle$ où c est une règle du langage sous-jacent et $P \subseteq T$ un ensemble non vide de règles défini tq :

1. P est consistant ;
2. $P \vdash c$;
3. P est minimal, c'est à dire qu'il n'existe pas de sous-ensemble de P qui vérifie 1 et 2.

P est la **prémisse** de A , notée $P = \text{premise}(A)$. c est la **conclusion** de A , notée $c = \text{conclusion}(A)$.

A' sera appelé **sous-argument** de A ssi la prémisse de A' est contenue dans la prémisse de A . Par exemple, $A' = \langle \{L \leftarrow\}, L \leftarrow \rangle$ est un sous-argument de $A = \langle \{L \leftarrow, M \leftarrow L\}, M \leftarrow \rangle$. De plus, nous dirons qu'un argument est **trivial** lorsque sa prémisse est constituée uniquement de sa conclusion. Par exemple, $A' = \langle \{L \leftarrow\}, L \leftarrow \rangle$ est un argument trivial.

Les agents partagent le même argumentaire, *i.e.* le même ensemble d'arguments construits sur la même théorie.

Définition n° 7.3.0.VIII [Argumentaire multi-agents]

Soit $\mathcal{AT}_{\cup_A} = \langle \mathcal{T}, V, \text{promote} \rangle$ une théorie argumentative multi-agents. L'**argumentaire multi-agents** est constitué de l'ensemble des arguments construits sur la théorie \mathcal{T} . On le note $\mathcal{A}(\mathcal{T})$.

La théorie n'est pas nécessairement consistante. Les inconsistances peut être explicites lorsque deux clauses unités de la théorie sont contradictoires (par exemple, $r : \text{president}(\text{jack}) \leftarrow$ et $r' : \neg \text{president}(\text{jack}) \leftarrow$). Les inconsistances peuvent être implicites lorsque les contradictions sont indirectes. L'exemple suivant illustre ce cas de figure.

Exemple n° 7.3.0.II [Arguments] On pourra noter que la théorie présentée dans l'exemple 7.2.0.1 n'est pas consistante. Elle permet de déduire $\text{president}(\text{jack})$ mais également $\text{president}(\text{jose})$. De plus, d'après les règles r_{11} et r_{12} , ces deux atomes s'excluent mutuellement. L'argumentaire multi-agents est constitué des arguments suivants :

1. D'une part, jack est actuellement président (r_6). D'autre part, le président actuel, jack , doit devenir le futur président ($r_3(\text{jack})$). On en déduit que jack doit devenir président :

$$A_1 = (\{r_6, r_3(\text{jack})\}, \text{president}(\text{jack}) \leftarrow)$$

2. *Le candidat jose est en prison (r_5). Or le futur président ne peut être en prison comme jose ($r_{22}(\text{jose})$). Si jose n'est pas le futur président alors jack doit devenir président (r_{11}). On en déduit que jack doit devenir président :*

$$A_2 = (\{r_5, r_{22}(\text{jose}), r_{11}\}, \text{president}(\text{jack}) \leftarrow)$$

L'argument A'_2 est un sous-argument de A_2 :

$$A'_2 = (\{r_5, r_{22}(\text{jose})\}, \neg\text{president}(\text{jose}) \leftarrow)$$

3. *Le candidat jack est un escroc (r_4). Or tous les escrocs comme jack vont en prison ($r_{12}(\text{jack})$). De plus, le futur président ne peut être en prison comme jack ($r_{22}(\text{jack})$). Si jack n'est pas le futur président alors jose doit devenir président (r_{21}). On en déduit que jose doit devenir président :*

$$B = (\{r_4, r_{12}(\text{jack}), r_{22}(\text{jack}), r_{21}\}, \text{president}(\text{jose}) \leftarrow)$$

L'argument B' est un sous-argument de B :

$$B' = (\{r_4, r_{12}(\text{jack}), r_{22}(\text{jack})\}, \neg\text{president}(\text{jack}) \leftarrow)$$

On dispose de deux arguments qui sont en faveur du candidat jack (A_1 et A_2) et d'une unique argument en faveur du candidat jose (B).

Parce que la théorie est inconsistante, les arguments peuvent avoir des conclusions contradictoires. Les relations entre arguments présentées dans la section suivante permettent de gérer de telles inconsistances.

7.4 Relations entre arguments

Dans un argumentaire multi-agents, les arguments sont liés les uns aux autres par des relations. Ces relations sont soit objectives soit subjectives. Elles sont objectives lorsque seules les relations de contradiction entre les règles sont considérées (cf section 7.4.1). Elles sont subjectives lorsqu'on considère également les inclinaisons des agents (cf section 7.4.2).

7.4.1 Relation d'attaque

Parce que la théorie peut être inconsistante, les arguments sont susceptibles d'être en conflit. La relation d'attaque présentée ici permet de capturer ces conflits. Nous considérons que deux arguments s'attaquent mutuellement s'ils ont des conclusions partielles contradictoires.

Définition n° 7.4.1.I [Attaque]

Soit $\mathcal{AT}_{\mathcal{U}_A} = \langle \mathcal{T}, V, \text{promote} \rangle$ une théorie argumentative multi-agents. Soient $A = \langle P, c \rangle$, $B = \langle P', c' \rangle \in \mathcal{A}(\mathcal{T})$ deux arguments. A **attaque** B (noté $\text{attacks}(A, B)$) ssi :

$$\exists L \in \mathcal{L} \exists P_1 \subseteq P, P_2 \subseteq P' \text{ tq } P_1 \vdash L \text{ et } P_2 \vdash \neg L$$

La relation d'attaque s'appuie sur la relation de conflit entre un atome et sa négation explicite. Cette relation est symétrique. Par abus de langage, on dit qu'un ensemble d'arguments S attaque A , ou bien A est attaqué par S (noté $\text{attacks}(S, A)$) ssi un argument de S attaque A .

Parce que les relations de priorité associées à chacun des agents diffèrent les unes des autres, ces attaques peuvent être ignorées.

7.4.2 Relations de défaite

Le succès d'une attaque d'un argument sur un autre argument dépend de l'agent qui l'évalue.

Comme cela est formalisé dans la définition ci-dessus, un argument défait pour un agent un autre argument dans la mesure où ils ont des conclusions partielles contradictoires et la prémisse du second sous-argument n'a pas un niveau de priorité supérieur.

Définition n° 7.4.2.I [Défaire pour un agent]

Soit $\mathcal{AT}_i = \langle \mathcal{T}, V, \text{promote}, \ll_i \rangle$ la théorie argumentative de l'agent ag_i . Soient $A = \langle P, c \rangle$, $B = \langle P', c' \rangle \in \mathcal{A}(\mathcal{T})$ deux arguments. A **défait** B **pour un agent** ag_i (noté $\text{defeats}_i(A, B)$) ssi $\exists L \in \mathcal{L} \exists P_1 \subseteq P, P_2 \subseteq P' \text{ tq} :$

1. $P_1 \vdash L$ et $P_2 \vdash \neg L$;
2. $\neg(\text{level}_i(P_1) \ll_i \text{level}_i(P_2))$.

Contrairement à la relation d'attaque, cette relation est asymétrique. Par abus de langage, on dit qu'un ensemble d'arguments S défait A pour un agent ag_i , ou bien A est défait par S (noté $\text{defeats}_i(S, A)$) ssi un argument de S défait A pour un agent ag_i .

Ainsi, le succès d'une attaque dépend de la **force des arguments**. La force d'un argument A , c'est à dire le niveau de priorité de sa prémisse, est évaluée indépendamment par chacun des agents : $\forall ag_i \in \mathcal{U}_A \text{ strength}_i(A) = \text{level}_i(\text{premise}(A))$.

L'exemple suivant permet d'instancier ces définitions.

Exemple n° 7.4.2.I [Relations entre arguments] Reprenons l'exemple précédent. Les arguments A_1 et A_2 permettent de déduire $\text{president}(\text{jack}) \leftarrow$. L'argument B permet de déduire $\text{president}(\text{jose}) \leftarrow$.

A_1 attaque B et réciproquement. En effet, l'argument B' est un sous-argument de B . De plus, sa conclusion est contradictoire avec celle de A_1 .

A_2 attaque B et réciproquement. En effet, l'argument A'_2 est un sous-argument de A_2 . De plus, sa conclusion est contradictoire avec celle de B .

Considérons dans un premier temps, le point de vue de l'agent ag_1 . L'ensemble des arguments sont représentés dans la figure 7.4.2.I à l'aide d'un graphe orienté dont les sommets sont des arguments et les arêtes représentent des relations de défaite. Les cellules grisées correspondent aux règles les moins prioritaires de chacune des prémisses.

Selon l'agent ag_1 , la valeur la moins prioritaire prônée par la prémisses de A_1 (en l'occurrence v_3) est moins prioritaire que la valeur prônée par la prémisses de B' (en l'occurrence v_4). En conséquence, B défait A_1 mais A_1 ne défait pas B . Selon l'agent ag_1 , la valeur la moins prioritaire prônée par la prémisses de A'_2 (en l'occurrence v_5) est plus prioritaire que la valeur prônée par la prémisses de B (en l'occurrence v_4). En conséquence, B ne défait pas A_2 mais A_2 défait B .

Considérons dans un second temps, le point de vue de l'agent ag_2 . L'ensemble des

\ll_1	V	premise(A_1)	premise(A_2)	premise(A'_2)	premise(B)	premise(B')
↑	v_1		r_{11}			
	v_2				r_{21}	
	v_3	$r_3(\text{jack})$				
	v_4				r_4	r_4
	v_5		r_5	r_5		
	v_6	r_6				

$A_2 \rightarrow B$
 $B \rightarrow A_1$

Fig. 7.4.2.I: Relations entre arguments selon l'agent ag_1

arguments sont représentés dans la figure 7.4.2.II.

Selon l'agent ag_2 , la valeur la moins prioritaire prônée par la prémisses de A_1 (en l'occurrence v_6) est moins prioritaire que la valeur prônée par la prémisses de B' (en l'occurrence v_4). En conséquence, B défait A_1 mais A_1 ne défait pas B . Selon l'agent ag_2 , la valeur la moins prioritaire prônée par la prémisses de A'_2 (en l'occurrence v_5) est moins prioritaire que la valeur prônée par la prémisses de B (en l'occurrence v_4). En conséquence, B défait A_2 mais A_2 ne défait pas B .

Ces relations ayant été définies, nous sommes alors en mesure de caractériser la position d'un argument dans une théorie argumentative.

\ll_2	V	premise(A_1)	premise(A_2)	premise(A'_2)	premise(B)	premise(B')
↑	v_1		r_{11}		r_{21}	
	v_2		$r_{22}(\text{jose})$	$r_{22}(\text{jack})$	$r_{12}(\text{jack})$ $r_{22}(\text{jack})$	$r_{12}(\text{jack})$ $r_{22}(\text{jack})$
	v_3	$r_3(\text{jack})$				
	v_4				r_4	r_4
	v_5		r_5	r_5		
	v_6	r_6				

$B \rightarrow A_1$
 $B \rightarrow A_2$

Fig. 7.4.2.II: Relation entre arguments selon l'agent ag_2

7.5 Acceptabilité des arguments

L'acceptabilité est une propriété des arguments qui permet de les catégoriser. Cette propriété dépend de l'agent qui l'évalue.

Considérons dans un premier temps, le point de vue d'un système multi-agents. L'ensemble des arguments objectivement acceptables est l'ensemble des arguments qui ne sont pas attaqués ou qui s'ils le sont, se défendent par eux-mêmes.

Définition n° 7.5.0.II [Acceptabilité objective]

Soit $\mathcal{AT}_{\mathcal{U}_A} = \langle \mathcal{T}, V, \text{promote} \rangle$ la théorie argumentative multi-agents. Soient $A \in \mathcal{A}(\mathcal{T})$ un argument objectif et $S \subseteq \mathcal{A}(\mathcal{T})$ un ensemble d'arguments. A est **objectivement acceptable vis à vis de S** ssi $\forall B \in \mathcal{A}(\mathcal{T}) \text{ attacks}(B, A) \Rightarrow \text{attacks}(S, B)$.

Les arguments considérés comme objectivement acceptables (ou non) sont dans l'argumentaire multi-agents. Ils sont évalués en fonction des relations objectives qui existent entre eux.

Considérons dans un second temps, le point de vue d'un agent. L'ensemble des arguments subjectivement acceptables est l'ensemble des arguments qui, selon lui, ne sont pas défaits ou qui s'ils le sont se défendent par eux-mêmes.

Définition n° 7.5.0.III [Acceptabilité subjective]

Soit $\mathcal{AT}_i = \langle \mathcal{T}, V, \text{promote}, \mathcal{U}_A, \ll_i \rangle$ la théorie argumentative de l'agent ag_i . Soient $A \in \mathcal{A}(\mathcal{T})$ un argument et $S \subseteq \mathcal{A}(\mathcal{T})$ un ensemble d'arguments. A est **subjectivement acceptable selon \mathcal{AT}_i vis à vis de S** ssi $\forall B \in \mathcal{A}(\mathcal{T}) \text{ defeats}_i(B, A) \Rightarrow \text{defeats}_i(S, B)$.

Les arguments considérés par un agent comme subjectivement acceptables (ou non) sont des arguments dans l'argumentaire multi-agents. Ils sont évalués en fonction des relations subjectives qui existent entre eux.

Par abus de langage, nous considérons dans la suite de ce manuscrit l'acceptabilité des arguments vis à vis de l'ensemble de l'argumentaire ($S = \mathcal{A}(\mathcal{T})$). Ainsi l'ensemble des arguments objectivement acceptables correspond à l'extension préférée du système d'argumentation $AS = \langle \mathcal{A}(\mathcal{T}), \text{attacks} \rangle$. De même l'ensemble des arguments subjectivement acceptables selon l'agent ag_i correspond à l'extension préférée du système d'argumentation $AS_i = \langle \mathcal{A}(\mathcal{T}), \text{defeats}_i \rangle$. Parce que le graphe associé à AS_i est sans circuit, l'extension préférée correspond à l'extension raisonnable. Elle est unique et non-vide (cf théorème 2.3.1.I). Contrairement à l'acceptabilité objective, l'acceptabilité subjective des arguments permet de conclure. Un agent est convaincu d'une conclusion s'il dispose d'un argument subjectivement acceptable en sa faveur.

L'exemple suivant illustre ces définitions.

Exemple n° 7.5.0.II [Acceptabilité] *Reprenons l'exemple précédent. La figure 7.5.0.III représente la théorie argumentative multi-agents et les deux théories argumentatives mono-agent précédemment évoquées. Les systèmes d'argumentation associés sont représentés à l'aide de graphes orientés. Considérons dans un premier temps la théorie*

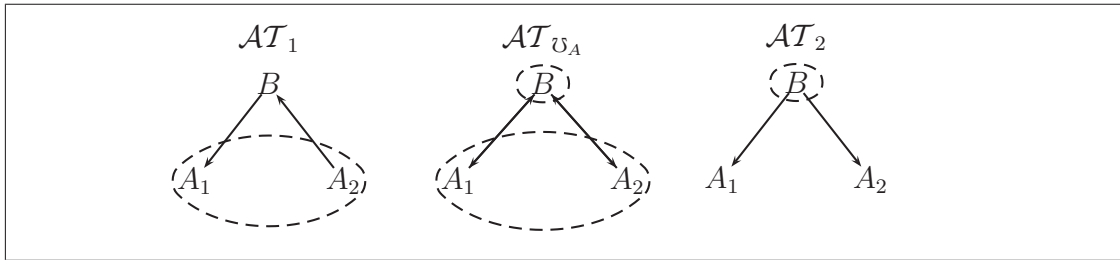


Fig. 7.5.0.III: Représentation des théories argumentatives AT_{\cup_A} (au centre), AT_1 (à gauche) et AT_2 (à droite)

argumentative multi-agents AT_{\cup_A} . Les ensembles $\{A_1A_2\}$ et $\{B\}$ sont objectivement acceptables.

Considérons dans un deuxième temps la théorie argumentative AT_1 . Contrairement à l'ensemble $\{B\}$, l'ensemble $\{A_1A_2\}$ est subjectivement acceptable.

Considérons dans un troisième temps la théorie argumentative AT_2 . Contrairement à l'ensemble $\{A_1, A_2\}$, l'ensemble $\{B\}$ est subjectivement acceptable.

Alors que l'agent ag_1 est convaincu que jack doit être président, l'agent ag_2 est convaincu que c'est jose qui doit être président. Il n'y a pas de consensus.

En résumé, l'acceptabilité des arguments est une propriété des arguments qui permet de les catégoriser. L'acceptabilité objective dépend uniquement des relations de contradiction entre les règles argumentatives. L'acceptabilité subjective dépend de

l'échelle de valeur à laquelle on se réfère et donc de l'agent qui l'évalue.

7.6 Synthèse

Nous avons proposé dans ce chapitre une logique argumentative qui constitue le modèle de raisonnement des agents. Cette logique argumentative permet de gérer les interactions entre des arguments contradictoires qui sont plus ou moins forts selon l'agent envisagé.

La logique argumentative proposée ici s'appuie sur un **langage logique**. Une théorie compile l'ensemble des connaissances communes aux agents. Chaque agent est associé à une relation de priorité qui explicite ses propres inclinaisons.

Les agents partagent des **arguments** qui sont construits à partir de leurs connaissances communes. Un argument est constitué d'une conclusion et d'une prémisses, i.e. un ensemble de règles à partir desquelles la conclusion est déduite.

Parce que ces connaissances sont contradictoires, les arguments sont susceptibles d'être en conflit. La **relation de contradiction** entre arguments, appelée ici attaque, permet de capturer ces conflits. Cette relation est objective.

Parce que les agents se réfèrent à des échelles de valeurs différentes, chacun d'eux évalue indépendamment la force des arguments. Chacune des **relations de priorité** permet de doter les arguments d'une force. Ainsi, le succès d'une attaque d'un argument sur un autre argument dépend de l'agent concerné.

Les relations entre arguments permettent d'aboutir à une catégorisation des arguments appelée **classes d'acceptabilité**. On distingue les arguments qui sont acceptés par le système multi-agents, et les arguments qui sont acceptés par un agent en particulier.

Dans ce chapitre, nous avons proposé une nouvelle logique argumentative. D'une part, la sémantique de cette logique argumentative est celle d'un système d'argumentation à base de valeurs [9]. Comme dans les travaux de Trevor Bench-Capon, plusieurs classes d'acceptabilité permettent de considérer simultanément le point de vue de plusieurs agents. D'autre part, la structure des arguments s'appuie sur un langage logique sous-jacent. Un argument est constitué d'une prémisses à partir de laquelle la conclusion est déduite. Comme dans les travaux d'Antonis Kakas et de Pavlos Moraitis [32, 42], les arguments sont construits à partir d'un programme logique.

Nous avons considéré jusqu'ici que les agents disposaient des mêmes connaissances. Ces connaissances étaient compilées dans une théorie commune à l'ensemble des

agents. Nous allons, dans le chapitre suivant, distinguer les connaissances propres à chacun des agents. Les éléments du modèle DIAL présentés dans le chapitre constituent un modèle informatique qui permet aux agents d'échanger leurs connaissances.

« Il serait salutaire de ne pas oublier que si les diverses parcelles de savoir que nous possédons nous rendent assez dissemblables, dans notre infinie ignorance nous sommes tous égaux. »

Karl R. Popper.

Chapitre 8

Systeme multi-agents argumentatifs

Sommaire

8.1	Introduction	137
8.2	Langages communs	138
8.2.1	Langage de représentation des connaissances	138
8.2.2	Langage de communication d'agents	138
8.3	Raisonnement conjoint	139
8.3.1	Agents argumentatifs	139
8.3.2	Théorie argumentative étendue	143
8.3.3	Sémantique sociale	145
8.3.4	Sémantique argumentative	148
8.4	Synthèse	152

8.1 Introduction

Ayant défini dans le chapitre précédent le modèle de raisonnement des agents, nous pouvons présenter maintenant les éléments du modèle DIAL qui permettent aux agents d'échanger des informations et de raisonner conjointement. Les agents font non seulement référence à des échelles de valeurs différentes mais ils ont également des connaissances différentes qu'ils échangent et à partir desquelles ils raisonnent.

Dans un premier temps, nous allons présenter dans la section 8.2 les deux langages partagés par l'ensemble des agents pour communiquer. Nous verrons ensuite dans la section 8.3 dans quelle mesure les agents raisonnent ensemble : comment les agents

sont constitués, comment ils interprètent les messages qui leur sont transmis et comment ils y répondent en fonction des arguments dont ils disposent.

8.2 Langages communs

Pour échanger des connaissances, les agents doivent se comprendre. Ils doivent partager d'une part un même langage de représentation des connaissances et d'autre part un langage de communication. Nous allons dans ce qui suit présenter ces deux langages.

8.2.1 Langage de représentation des connaissances

Un système multi-agents argumentatifs est constitué d'un ensemble d'agents (noté \mathcal{U}_A) parmi lesquels la connaissance du système est distribuée. Ces connaissances sont des expressions bien formées d'un même **langage de représentation des connaissances** (noté $\mathcal{L}_{\mathcal{U}}$). Afin que les agents tirent, à partir des mêmes connaissances, les mêmes conclusions, ils partagent le même mécanisme d'inférence (noté $\vdash_{\mathcal{U}}$). Dans la suite de ce manuscrit, nous parlerons indifféremment d'agent ou d'agent argumentatif.

Pour échanger des connaissances, un langage de communication d'agents est nécessaire.

8.2.2 Langage de communication d'agents

Les agents émettent des messages pour échanger leurs connaissances. À cet effet, nous définissons un **langage de communication d'agents** (noté $\mathcal{C}\mathcal{L}_{\mathcal{U}}$). Chaque message est muni d'un identifiant unique $M_k \in \mathcal{C}\mathcal{L}_{\mathcal{U}}$.

Définition n° 8.2.2.I [Message]

Un **message** $M_k \in \mathcal{C}\mathcal{L}_{\mathcal{U}}$ est défini par un triplet $M_k = \langle S_k, H_k, A_k \rangle$ où :

- $S_k = \text{speaker}(M_k) \in \mathcal{U}_A$ est le locuteur du message ;
- $H_k = \text{hearer}(M_k) \in \mathcal{U}_A$ en est l'allocutaire ;
- $A_k = \text{act}(M_k)$ correspond à l'acte de langage associé au message. Il est composé d'une locution et d'un contenu. La locution est l'une des suivantes : **question**, **assert**, **unknow**, **concede**, **challenge**, **withdraw**. Le contenu, appelé **hypothèse**, est constitué d'une règle, de la négation d'une règle ou d'un ensemble de règles issues du langage $\mathcal{L}_{\mathcal{U}}$.

Un message est émis par un locuteur à l'intention d'un allocataire. Tous les autres agents du système sont de simples auditeurs du message. Seul l'allocataire d'un message l'interprète dans le but d'y répondre.

Ayant défini les deux langages communs aux agents, nous allons dans la section suivante décrire les agents et voir comment ils raisonnent de manière conjointe.

8.3 Raisonnement conjoint

Chaque agent est muni de ses propres croyances. Il est associé à une théorie argumentative étendue. Cette théorie argumentative (cf section 7.2) permet de prendre en considération les hypothèses de ses interlocuteurs. D'une part, un agent interprète les hypothèses qui lui sont transmises et les évalue sur la base de la compétence estimée de l'agent qui a transmis cette information. D'autre part, l'agent émet des hypothèses. On dit des agents qu'ils **co-argumentent**.

Nous allons tout d'abord définir les éléments constitutifs de chacun des agents argumentatifs (cf section 8.3.1). À partir de ces éléments, nous construisons les théories argumentatives étendues (cf section 8.3.2). Les hypothèses reçues venant enrichir la théorie argumentative étendue, les actes de langages ont une sémantique sociale (cf section 8.3.3). Les hypothèses émises pouvant être justifiées par la théorie argumentative étendue, les actes de langage ont une sémantique argumentative (cf section 8.3.4).

8.3.1 Agents argumentatifs

Dans un système multi-agents argumentatifs, les connaissances mais également les inclinaisons des agents diffèrent. Ils partagent toutefois des valeurs et des connaissances communes. De plus, ils enregistrent les hypothèses émises par leurs interlocuteurs.

Les agents argumentatifs, qui disposent de leurs croyances et de leurs valeurs, enregistrent les engagements pris par leurs interlocuteurs dont ils évaluent individuellement la réputation. Chacun des agents est conforme à la définition suivante :

Définition n° 8.3.1.I [Agent argumentatif]

Un agent argumentatif $ag_i \in \mathcal{U}_A$ est défini par un sextuplet

$ag_i = \langle \mathcal{T}_i, V_i, \ll_i, promote_i, \cup_{j \neq i} CS_j^i(t), \prec_i \rangle$ où :

- $\mathcal{T}_i \subseteq \mathcal{T}_{\mathcal{U}_A}$ est la **théorie personnelle** de l'agent, i.e. un ensemble de règles personnelles qui constituent les croyances de l'agent ;
- $V_i \subseteq V_{\mathcal{U}_A}$ est un ensemble fini de **valeurs personnelles** qui constituent les valeurs considérées par l'agent ;
- $promote_i : \mathcal{T}_i \rightarrow V_i$ est une fonction qui met en relation les règles personnelles et les valeurs personnelles ;
- \ll_i est la relation de priorité associée à l'agent, i.e. une relation d'ordre strict et total sur V_i ;
- $CS_j^i(t)$ est le tableau d'engagement constitué des engagements pris avant ou à l'instant t dont l'agent ag_i est le créateur et l'agent ag_j le débiteur ;
- \prec_i est la **relation de réputation** associée à l'agent, i.e. une relation d'ordre strict et total sur \mathcal{U}_A ;

Les connaissances des agents diffèrent, on parle alors de croyances. Chaque agent a sa propre théorie personnelle \mathcal{T}_i . La multiplicité des théories personnelles justifie l'emploi de cette notion épistémique. Ces théories ne sont pas nécessairement disjointes. Nous appellerons **théorie commune** (noté \mathcal{T}_{Ω_A}) les règles explicitement partagées par les agents argumentatifs :

$$\mathcal{T}_{\Omega_A} \subseteq \cap_{ag_i \in \mathcal{U}_A} \mathcal{T}_i$$

L'ensemble des connaissances sont distribuées parmi les agents argumentatifs du système. Elles peuvent être complémentaires ou contradictoires. Nous appellerons **théorie conjointe** (noté $\mathcal{T}_{\mathcal{U}_A}$) l'ensemble des règles du système :

$$\mathcal{T}_{\mathcal{U}_A} = \cup_{ag_i \in \mathcal{U}_A} \mathcal{T}_i$$

Les agents ne se réfèrent pas nécessairement aux mêmes valeurs. Chaque agent dispose d'un ensemble de valeurs personnelles V_i . Ces ensembles ne sont pas nécessairement disjoints. Nous appellerons **valeurs communes** (noté V_{Ω_A}), les valeurs explicitement partagées par les agents argumentatifs :

$$V_{\Omega_A} \subseteq \cap_{ag_i \in \mathcal{U}_A} V_i$$

Une théorie personnelle (\mathcal{T}_i) fait référence à un ensemble de valeurs personnelles (V_i). La fonction $promote_i : \mathcal{T}_i \rightarrow V_i$ met en relation la théorie personnelle avec les valeurs personnelles. Chacune des règles argumentatives prône une valeur :

$$\forall r \in \mathcal{T}_i \exists v \in V_i \text{ promote}_i(r) = v$$

D'une part, la théorie commune (\mathcal{T}_{Ω_A}) fait référence à l'ensemble de valeurs communes (V_{Ω_A}). La fonction $\text{promote}_{\Omega_A} : \mathcal{T}_{\Omega_A} \rightarrow V_{\Omega_A}$ met en relation ces deux ensembles. Chacune des règles communes prône une valeur commune :

$$\forall r \in \mathcal{T}_{\Omega_A} \exists v \in V_{\Omega_A} \text{promote}_{\Omega_A}(r) = v$$

D'autre part, une **théorie propre** à l'agent ($\mathcal{T}_i - \mathcal{T}_{\Omega_A}$) fait référence à un ensemble de **valeurs propres** à l'agent ($V_i - V_{\Omega_A}$). Chacune des règles propres à un agent prône une valeur qui lui est propre :

$$\forall r \in \mathcal{T}_i - \mathcal{T}_{\Omega_A} \exists v \in V_i - V_{\Omega_A} \text{promote}_i(r) = v$$

Quand elles sont communes, les valeurs sont universelles et quasi-semblables à des vérités. Quand elles sont propres à l'agent, les valeurs se présentent simplement comme conformes aux aspirations de l'agent [15] (cf section 2.2.3). En d'autres termes, les valeurs communes sont prioritaires vis à vis des valeurs propres à l'agent :

$$\forall v_\omega \in V_{\Omega_A} \forall v \in V_i - V_{\Omega_A} v \ll_i v_\omega$$

Ainsi, les agents argumentatifs du système partagent la théorie argumentative multi-agents suivante :

$$\mathcal{AT}_{\Omega_A} = \langle \mathcal{T}_{\Omega_A}, V_{\Omega_A}, \text{promote}_{\Omega_A} \rangle$$

Au cours des échanges, les agents prennent position. Les auditeurs de ces messages enregistrent ces positions. On nomme tableaux d'engagements (*commitment stores*) les structures de données qui font le compte des déclarations perçues par un agent et émises par ses interlocuteurs au cours des échanges. Le tableau d'engagement, noté CS_j^i , répertorie l'ensemble des règles sur lesquelles l'agent ag_j s'engage et qui sont enregistrées par l'agent ag_i . Initialement vides, les tableaux d'engagement conservent la trace des engagements au cours des échanges.

La réputation est un concept social qui lie un agent à ses interlocuteurs. C'est également une relation nivelée [16]. Les relations individuelles de réputation qui sont transitives, irreflexives, asymétriques et totales, vérifient ces hypothèses. $\text{ag}_j \prec_i \text{ag}_k$ se lit : « l'agent ag_i fait plus confiance à l'agent ag_k qu'à l'agent ag_j ».

L'exemple suivant permet d'illustrer cette définition.

Exemple n° 8.3.1.I [Agent argumentatif] *Reprenons l'exemple 7.2.0.I du chapitre précédent. Les adhérents d'une association loi 1901 doivent, au cours d'une assemblée générale, se prononcer pour désigner le futur président de cette association. Considérons deux agents : ag_1 et ag_2 .*

L'agent ag_1 est l'un de ces adhérents. Sa théorie personnelle (notée \mathcal{T}_1) est constituée de

cinq règles qui prônent trois valeurs différentes. Cet agent peut être associé à la théorie argumentative mono-agent $\mathcal{AT}_1 = \langle \mathcal{T}_1, V_1, \text{promote}_1, \ll_1 \rangle$ représentée dans la figure 8.3.1.I. Dans cette figure, une valeur située au-dessus d'une autre valeur est considérée par l'agent comme davantage prioritaire. On vérifie que : $v_3 \ll_1 v_2 \ll_1 v_1$.

L'agent ag_2 est un second adhérent. Sa théorie personnelle (notée \mathcal{T}_2) est constituée de huit règles qui prônent six valeurs différentes. Cet agent peut être associé à la théorie argumentative mono-agent $\mathcal{AT}_2 = \langle \mathcal{T}_2, V_2, \text{promote}_2, \ll_2 \rangle$ représentée dans la figure 8.3.1.II.

\ll_1	V_1	\mathcal{T}_1	\mathcal{T}_{Ω_A}
↑	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	
	v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$	
	v_3	$r_3(x) : \text{president}(x) \leftarrow \text{president_actuel}(x)$	

Fig. 8.3.1.I: Théorie argumentative associée à l'agent ag_1

\ll_2	V_2	\mathcal{T}_2	\mathcal{T}_{Ω_A}
↑	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	
	v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$	
	v_3	$r_3(x) : \text{president}(x) \leftarrow \text{president_actuel}(x)$	
	v_4	$r_4 : \text{escroc}(\text{jack}) \leftarrow$	
	v_5	$r_5 : \text{prison}(\text{jose}) \leftarrow$	
	v_6	$r_6 : \text{president_actuel}(\text{jack}) \leftarrow$	

Fig. 8.3.1.II: Théorie argumentative associée à l'agent ag_2

En résumé, ces deux agents utilisent un même langage de représentation de connaissances. Ils partagent explicitement une théorie commune. Celle-ci est constituée des règles relatives à l'objectif et des règles de sens commun ($\mathcal{T}_{\Omega_A} = \{r_{11}, r_{21}, r_{12}(x), r_{22}(x)\}$). Ces règles prônent un ensemble de valeurs communes ($V_{\Omega_A} = \{v_1, v_2\}$).

Contrairement au modèle AMP [4, 46], les hypothèses notifiées dans les tableaux d'engagements sont prises en considération dans le processus de raisonnement des agents. Nous allons dans la section suivante présenter les théories argumentatives étendues associées à chacun des agents qui permettent de relever ce défi.

8.3.2 Théorie argumentative étendue

Puisque les connaissances sont distribuées parmi les agents du système, chaque agent dispose d'un argumentaire qui lui est propre. Une règle peut être : soit commune à l'ensemble des agents, soit une croyance propre à l'agent, soit une déclaration émise par l'un de ses interlocuteurs. La priorité de cette règle dépend alors de la source de l'information. Nous associons à chaque agent une théorie argumentative étendue qui associe des valeurs différentes à chacune des sources d'information.

Comme chaque agent ag_i dispose d'une théorie argumentative personnelle (\mathcal{T}_i), d'un ensemble de valeurs personnelles (V_i) et d'une relation de priorité (\ll_i), nous pouvons trivialement l'associer à la théorie argumentative mono-agent suivante :

$$\mathcal{AT}_i = \langle \mathcal{T}_i, V_i, \text{promote}_i, \ll_i \rangle$$

De cette manière, l'argumentaire d'un agent est construit seulement à partir de ses croyances personnelles. C'est la solution adoptée par le modèle AMP [46, 4]. Une telle théorie argumentative ne permet pas aux agents de raisonner ensemble. À cet effet, les agents doivent construire leur argumentaire non seulement à partir de leurs croyances mais également à partir des engagements de leurs interlocuteurs. Par conséquence, on associe à chaque agent la théorie argumentative étendue suivante :

Définition n° 8.3.2.I [Théorie argumentative étendue]

Soit $ag_i \in \mathcal{U}_A$ un agent argumentatif. La **théorie argumentative étendue associée à l'agent ag_i** est un quadruplet $\mathcal{AT}_i^* = \langle \mathcal{T}_i^*, V_i^*, \text{promote}_i^*, \ll_i^* \rangle$ où :

- $\mathcal{T}_i^* = \mathcal{T}_i \cup [\bigcup_{j \neq i} CS_j^i]$ est la **théorie personnelle étendue** associée à l'agent ag_i qui comprend la théorie personnelle de cet agent et l'ensemble des hypothèses émises par ses interlocuteurs ;
- $V_i^* = V_i \cup [\bigcup_{j \neq i} \{v_j^i\}]$ est l'**ensemble étendu des valeurs personnelles** associées à l'agent ag_i qui est constitué de l'ensemble des valeurs personnelles (V_i) et d'un ensemble de **valeurs de réputation** ($\bigcup_{j \neq i} \{v_j^i\}$). Chaque valeur v_j^i est associée à la perception de l'agent ag_j par l'agent ag_i ;
- $\text{promote}_i^* : \mathcal{T}_i^* \rightarrow V_i^*$ est l'extension de la fonction promote_i qui à chacune des règles de la théorie personnelle étendue (\mathcal{T}_i^*) associe une des valeurs de V_i^* . D'une part, les règles de \mathcal{T}_i sont, conformément à promote_i , associées aux valeurs de V_i . D'autre part, les règles du tableau d'engagement CS_j^i sont associées à la valeur v_j^i ;
- \ll_i^* est la **relation étendue de priorité** associée à l'agent ag_i , i.e. une relation d'ordre strict et total sur V_i^* .

Les agents peuvent faire des choix différents selon l'échelle de valeurs à laquelle ils se réfèrent. À cet effet, l'ensemble des valeurs personnelles (V_i) est muni d'une relation

d'ordre strict et total \ll_i . De même, l'ensemble étendu des valeurs personnelles (V_i^*) est muni d'une relation d'ordre strict et total (notée \ll_i^*). Cette seconde relation est le prolongement de la première relation. Bien évidemment, la relation étendue de priorité conserve la même hiérarchie sur les valeurs personnelles :

$$\forall v_1, v_2 \in V_i \ (v_1 \ll_i v_2 \Rightarrow v_1 \ll_i^* v_2)$$

Étant universelles et quasi-semblables à des vérités, les valeurs communes sont prioritaires sur les autres valeurs :

$$\forall v_\omega \in V_{\Omega_A} \forall v \in V_i^* - V_{\Omega_A} \ v \ll_i^* v_\omega$$

La relation étendue de priorité d'un agent permet de comparer les valeurs de réputation de ses interlocuteurs. Nous reviendrons sur ce point dans la section 8.3.3.

Ayant défini la théorie argumentative étendue d'un agent, ce dernier peut catégoriser les arguments. On dit qu'un argument est acceptable par un agent argumentatif si cet argument est, selon la théorie argumentative étendue, acceptable vis à vis de l'ensemble de l'argumentaire.

Définition n° 8.3.2.II [Acceptabilité d'un agent argumentatif]

Soit $ag_i \in \mathcal{U}_A$ un agent argumentatif qui est associé à la théorie argumentative étendue $\mathcal{AT}_i^* = \langle \mathcal{T}_i^*, V_i^*, \text{promote}_i^*, \ll_i^* \rangle$. Soit $A \in \mathcal{A}(\mathcal{T}_i^*)$ un argument. A est **acceptable selon l'agent argumentatif** ag_i ssi A est subjectivement acceptable selon \mathcal{AT}_i^* vis à vis de $\mathcal{A}(\mathcal{T}_i^*)$.

L'ensemble des arguments subjectivement acceptables selon l'agent argumentatif ag_i sera noté \mathcal{S}_i^* . On dit d'un agent argumentatif qu'il est **convaincu** par une règle s'il dispose d'un argument subjectivement acceptable en sa faveur.

L'exemple suivant permet d'illustrer ces définitions.

Exemple n° 8.3.2.I [Théorie argumentative étendue] L'agent ag_2 peut être associé à la théorie argumentative étendue suivante :

$$\mathcal{AT}_2^* = \langle \mathcal{T}_2^*, V_2^*, \text{promote}_2^*, \ll_2^* \rangle$$

Cette théorie mono-agent est représentée dans la figure 8.3.2.1. La théorie personnelle étendue est constituée d'une part de la théorie personnelle (r_{11}, \dots, r_6) et d'autre part de l'hypothèse émise par son interlocuteur (r_7). L'ensemble des valeurs personnelles est constitué d'une part de ses valeurs personnelles ($\{v_1, \dots, v_6\}$) et d'autre part de la valeur de réputation de l'agent ag_1 (v_1^2). Les règles communes ($r_{11}, r_{21}, r_{12}, r_{22}$) sont associées aux valeurs communes (v_1, v_2). Les règles propres à l'agent (r_3, \dots, r_6) sont associées

\ll_2^*	V_2^*	T_2^*	\mathcal{T}_{Ω_A}
↑	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	
	v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$	
	v_1^2	$\text{CS}_1^2 \supseteq r_7 : \text{president}(\text{jack}) \leftarrow$	
	v_3	$r_3(x) : \text{president}(x) \leftarrow \text{president_actuel}(x)$	
	v_4	$r_4 : \text{escroc}(\text{jack}) \leftarrow$	
	v_5	$r_5 : \text{prison}(\text{jose}) \leftarrow$	
v_6	$r_6 : \text{president_actuel}(\text{jack}) \leftarrow$		

Fig. 8.3.2.I: Théorie argumentative étendue associée à l'agent ag_2

aux valeurs propres v_3, \dots, v_6 . La règle r_7 , dont l'agent ag_1 est la source, est associée à la valeur v_1^2 . L'agent ag_2 dispose d'arguments en faveur de chacun des candidats. Les arguments A' , A_1 et A_2 sont en faveur du candidat jack. L'argument B est en faveur du candidat jose. Les relations subjectives entre les arguments sont représentées à l'aide d'un graphe orienté dans la figure 8.3.2.I. L'ensemble $\{A', A_1, A_2\}$ est l'ensemble des arguments acceptables selon l'agent argumentatif ag_2 . Ce dernier est convaincu que jack doit devenir président.

En résumé, un agent est associé à une théorie argumentative étendue afin d'argumenter avec ses interlocuteurs. Les hypothèses reçues sont interprétées, elles viennent enrichir son argumentaire. En d'autres termes, les actes de langage ont une sémantique sociale.

8.3.3 Sémantique sociale

La sémantique sociale des actes de langage permet de les interpréter. Selon l'acte reçu, les tableaux d'engagement sont (ou non) mis à jour. Les hypothèses ainsi adoptées doivent être évaluées. Leur priorité dépend de la compétence estimée de l'agent qui a transmis l'hypothèse.

8.3.3.1 Engagements sociaux

Les agents argumentatifs prennent position au cours des échanges. Le sens des actes de langage est défini par la notion d'engagement social, c'est à dire un engagement qui lie le locuteur à un auditeur vis à vis d'une hypothèse. Cette notion est formalisée à l'aide des tableaux d'engagement.

Dans la section 8.3.1, nous avons associé à chaque agent ag_i un ensemble de tableaux d'engagement $\cup_{j \neq i} CS_j^i$ qui contient les positions prises par ses interlocuteurs ag_j (avec $j \neq i$) au cours des échanges. Les tableaux d'engagement sont mis à jour selon les règles suivantes :

Définition n° 8.3.3.I [Règle de mise à jour des engagements]

Soit $M_k \in \mathcal{CL}_{\mathcal{U}}$, un message émis à l'instant t par l'agent ag_j et dont l'agent ag_i est l'un des auditeurs.

- si $act(M_k) = question(H)$ ou $act(M_k) = unknow(H)$
ou $act(M_k) = challenge(H)$ ou $act(M_k) = withdraw(H)$ alors
 $CS_j^i(t) = CS_j^i(t - 1)$;
- si $act(M_k) = assert(H)$ ou $act(M_k) = concede(H)$ alors
 $CS_j^i(t) = CS_j^i(t - 1) \cup H$.

On peut remarquer que les tableaux d'engagement sont cumulatifs, *i.e.* la mise à jour se limite à l'ajout de nouveaux engagements. De plus, les hypothèses reçues sont toutes supportées par un argument trivial dans la théorie argumentative étendue. Nous reviendrons sur ce point dans la section 8.3.4.1.

Ainsi, les agents argumentatifs prennent en considération les hypothèses de leurs interlocuteurs. En effet, les hypothèses reçues sont adoptées. Toutefois, leur priorité doit être évaluée.

8.3.3.2 Principe coopératif d'adoption des hypothèses

Les relation de priorités étant statiques, la priorité d'une hypothèse adoptée au cours des échanges doit être évaluée *a priori*. À cet effet, le processus le plus raffiné consiste à évaluer la priorité d'une hypothèse sur la base de la compétence estimée de l'agent qui est à l'origine de l'information. Chaque agent ag_i estime la réputation des agents du système à l'aide d'une relation d'ordre total et strict.

Les priorités entre les règles sont évaluées selon le **principe coopératif d'adoption des hypothèses** suivant : « *L'agent ag_i préfère les engagements pris par l'agent ag_j si et seulement si ag_j est considéré comme plus compétent* » [27]. Ce principe de sens commun est formalisé dans la définition suivante :

Définition n° 8.3.3.II [Compétence estimée]

Soient $ag_i, ag_j, ag_k \in \mathcal{U}_A$, trois agents.

- **L'agent ag_i considère l'agent ag_j comme plus (resp moins) compétent que lui**, noté $ag_i \prec_i ag_j$ (resp $ag_j \prec_i ag_i$) ssi :

$$\forall v \in V_i - V_{\Omega_A} \quad v \ll_i^* v_j^i \text{ (resp } v_j^i \ll_i^* v)$$

- **L'agent ag_i considère l'agent ag_j comme plus (resp moins) compétent que l'agent ag_k** , noté $ag_k \prec_i ag_j$ (respectivement $ag_j \prec_i ag_k$) ssi :

$$v_k^i \ll_i^* v_j^i \text{ (resp } v_j^i \ll_i^* v_k^i)$$

À partir de cette définition, nous sommes maintenant en mesure de valuer différemment les règles selon si elles sont issues de la théorie argumentative personnelle (\mathcal{T}_i) ou si elles sont issues des différents tableaux d'engagement (CS_j^i, CS_k^i, \dots). La relation de réputation étant une relation d'ordre strict et total, la relation de priorité est également une relation d'ordre strict et total sur V_i^* .

L'exemple suivant permet d'illustrer cette définition.

Exemple n° 8.3.3.I [Sémantique sociale] *Lorsqu'un agent ag_3 reçoit le message suivant :*

$$M_1 = \langle ag_1, ag_2, \text{assert}(\text{president}(\text{jose}) \leftarrow) \rangle$$

le tableau d'engagement correspondant, initialement vide, est mis à jour :

$$CS_1^3 = \{ \text{president}(\text{jose}) \leftarrow \}$$

Lorsque l'agent ag_3 reçoit le message suivant :

$$M_2 = \langle ag_2, ag_1, \text{assert}(\text{president}(\text{jack}) \leftarrow) \rangle$$

le tableau d'engagement correspondant, initialement vide, est mis à jour :

$$CS_2^3 = \{ \text{president}(\text{jack}) \leftarrow \}$$

L'agent ag_3 considère l'agent ag_2 plus compétent que l'agent ag_1 : $ag_1 \prec_3 ag_2$. La valeur v_1^3 associée au tableau d'engagement CS_1^3 est moins prioritaire que la valeur v_2^3 associée au tableau d'engagement CS_2^3 : $v_1^3 \ll_3^ v_2^3$.*

En résumé, l'argumentaire d'un agent est contruit à partir de ses croyances et des engagements pris par ses interlocuteurs. Comme nous allons le voir dans la section suivante, cet argumentaire permet de justifier les hypothèses sur lesquelles l'agent s'engage.

8.3.4 Sémantique argumentative

La sémantique argumentative des actes de langage permet de les énoncer. La condition rationnelle d'énonciation d'un acte de langage détermine dans quelle mesure un agent peut émettre un acte. Nous spécifions les conditions rationnelles d'énonciation des actes de langage à partir de la théorie argumentative étendue du locuteur (cf section 8.3.4.1). Les conditions rationnelles d'énonciation des actes sont communes à l'ensemble des agents. Parce que les conditions rationnelles d'énonciation de deux actes ne sont pas nécessairement mutuellement exclusives, les agents réalisent des choix individuels. Nous définissons les tactiques d'énonciation que l'agent adopte pour réaliser de tels choix (cf section 8.3.4.2). Contrairement aux conditions d'énonciation, les tactiques d'énonciation dépendent de l'attitude de l'agent considéré.

8.3.4.1 Conditions rationnelles d'énonciation

La condition rationnelle d'énonciation d'un acte de langage est fonction du contenu et dépend également de la locution utilisée.

Nous considérons qu'un agent peut affirmer une hypothèse s'il possède un argument en sa faveur.

Définition n° 8.3.4.I [Condition d'énonciation d'une affirmation]

On note $can_assert(ag_i, H)$ le prédicat défini tel que :

$$can_assert(ag_i, H) \Leftrightarrow \forall h \in H \exists A \in \mathcal{A}(T_i^*) \text{ conclusion}(A) = h$$

$can_assert(ag_i, H)$ est appelé **condition rationnelle d'énonciation de l'affirmation d'une hypothèse H par l'agent ag_i** .

Contrairement au modèle AMP [4, 46], la condition rationnelle d'énonciation d'une affirmation et celle d'une concession diffèrent.

Définition n° 8.3.4.II [Condition d'énonciation d'une concession]

On note $can_concede(ag_i, H)$ le prédicat défini tel que :

$$can_concede(ag_i, H) \Leftrightarrow \forall h \in H \exists A \in \mathcal{A}(T_i^*) \text{ conclusion}(A) = h \text{ avec} \\ \text{conclusion}(A) \not\subseteq \text{premise}(A) \wedge \text{premise}(A) \not\subseteq \cup_{j \neq i} CS_j^i$$

$can_concede(ag_i, H)$ est appelé **condition rationnelle d'énonciation de la concession de l'hypothèse H par un agent ag_i** .

Les agents peuvent affirmer une hypothèse qu'elle soit supportée par un argument trivial ou non. À l'inverse, les agents peuvent concéder une hypothèse s'ils disposent d'un argument personnel et non-trivial en sa faveur. Un argument est personnel s'il s'appuie sur les propres croyances de l'agent. En d'autres termes, les agents ne peuvent pas concéder toutes les hypothèses qui leur sont soumises, bien qu'elles soient toutes supportées par un argument trivial dans sa théorie argumentative étendue. Nous verrons dans le chapitre suivant que la condition rationnelle d'énonciation n'est pas le seul élément du modèle DIAL qui distingue une affirmation d'une concession.

Dans le but de simplifier la notation des actes de langage, $\text{concede}(\neg L \leftarrow)$ est noté $\text{refuse}(L \leftarrow)$ et $\text{assert}(\neg L \leftarrow)$ est noté $\text{reject}(L \leftarrow)$.

Puis que les autres actes de langage ($\text{question}(H)$, $\text{challenge}(H)$, $\text{unknow}(H)$, $\text{withdraw}(H)$) n'ont pas d'effet sur les tableaux d'engagements, ils n'ont pas de condition rationnelle d'énonciation particulière.

Définition n° 8.3.4.III [Autres conditions d'énonciation]

On note $\text{can_question}(ag_i, H)$, $\text{can_unknow}(ag_i, H)$, $\text{can_challenge}(ag_i, H)$ et $\text{can_withdraw}(ag_i, H)$, les prédicats définis de la manière suivante :

$$\begin{aligned} \text{can_question}(ag_i, H) &\Leftrightarrow \text{can_unknow}(ag_i, H) \Leftrightarrow \\ \text{can_challenge}(ag_i, H) &\Leftrightarrow \text{can_withdraw}(ag_i, H) \Leftrightarrow \text{true} \end{aligned}$$

On dit que $\text{can_question}(ag_i, H)$, $\text{can_unknow}(ag_i, H)$, $\text{can_challenge}(ag_i, H)$ et $\text{can_withdraw}(ag_i, H)$, sont les **conditions rationnelles d'énonciation** respectives **d'une question, d'un aveu d'ignorance, d'une mise en doute et d'un abandon d'une hypothèse H par l'agent ag_i .**

L'exemple suivant permet d'illustrer ces définitions.

Exemple n° 8.3.4.I [Conditions d'énonciation] *Considérons l'agent ag_2 tel que nous l'avons défini précédemment. L'agent ag_2 dispose d'arguments en faveur de chacun des candidats. Les arguments A_1 et A_2 sont en faveur du candidat jack. L'argument B est en faveur du candidat jose. Supposons que l'agent ag_1 ait pris position en faveur du candidat jack ($\text{president}(\text{jack}) \leftarrow \subseteq CS_1^2$). L'agent ag_2 dispose alors d'un nouvel argument :*

$$A' = (\{\text{president}(\text{jack}) \leftarrow\}, \text{president}(\text{jack}) \leftarrow)$$

Cet argument est trivial ($\text{premise}(A') \subseteq \text{conclusion}(A')$). De plus, il s'appuie sur des règles issues du tableau d'engagement ($\text{premise}(A) \subseteq CS_1^2$). Considérant ce seul argument, l'hypothèse $\text{president}(\text{jack}) \leftarrow$ ne pourrait pas, d'après la définition 8.3.4.II, être concédée par l'agent ag_2 . Toutefois, l'hypothèse $\text{president}(\text{jack}) \leftarrow$ peut être concédée

en vertu de l'argument A_1 ou de l'argument A_2 . De même, l'hypothèse $\text{president}(\text{jose}) \leftarrow$ peut être concédée en vertu de l'argument B .

Étant donné que l'agent ag_2 dispose d'arguments en faveur de chacun des candidats, les hypothèses $\text{president}(\text{jose}) \leftarrow$ et $\text{president}(\text{jack}) \leftarrow$ peuvent être affirmées toutes deux. Rien ne permet jusqu'ici à l'agent ag_2 de choisir laquelle de ces deux hypothèses doit être affirmée.

En résumé, les conditions rationnelles d'énonciation, qui détermine si un acte peut être émis, sont des prédicats dont les définitions ne diffèrent pas d'un agent à un autre. À l'inverse, les tactiques d'énonciations révèlent des choix individuels de l'agent locuteur.

8.3.4.2 Tactiques d'énonciation

Comme nous l'avons vu dans l'exemple précédent, les conditions d'énonciation de deux actes ne sont pas nécessairement mutuellement exclusives. Un agent peut, *via* sa théorie argumentative étendue, disposer simultanément d'un argument en faveur d'une hypothèse et d'un argument en faveur de la négation de cette hypothèse. De telles situations d'indéterminisme rendent possible un choix personnel de l'agent. C'est la raison pour laquelle nous définissons un ensemble de tactiques argumentatives. Chacune de ces tactiques restreint les conditions d'énonciation d'une hypothèse.

Dans un premier temps, un agent est qualifié de prévenant lorsqu'il souhaite affirmer une hypothèse pour laquelle il dispose d'un argument acceptable. Sans cette restriction, l'agent est confiant.

Définition n° 8.3.4.IV [Tactiques d'énonciation d'une affirmation]

On note $\text{want_assert}(ag_i, H)$ le prédicat défini tel que :

– si ag_i est **prévenant** (noté $\text{thoughtful}(ag_i)$) alors

$$\text{want_assert}(ag_i, H) \Leftrightarrow \forall h \in H \exists A \in \mathcal{S}_i^* \text{ conclusion}(A) = h$$

– si ag_i est **confiant** ($\text{confident}(ag_i)$) alors :

$$\text{want_assert}(ag_i, H) \Leftrightarrow \text{can_assert}(ag_i, H)$$

On dit que $\text{want_assert}(ag_i, H)$ est la **tactique d'énonciation de l'affirmation d'une hypothèse H par l'agent ag_i** . Elle dépend de l'attitude argumentative de l'agent locuteur : prévenant ou confiant.

Pour la même raison que celle évoquée dans la section 8.3.4.1, la tactique d'énon-

ciation d'une concession se distingue de la tactique d'énonciation d'une affirmation. Toutefois, elles sont définies de manière très similaire.

Définition n° 8.3.4.V [Tactiques d'énonciation d'une concession]

On note $want_concede(ag_i, H)$ le prédicat défini tel que :

– si ag_i est **sceptique** (noté $skeptical(ag_i)$) alors

$$want_concede(ag_i, H) \Leftrightarrow \forall h \in H \exists A \in \mathcal{S}_i^* \text{ conclusion}(A) = h \text{ avec} \\ \text{conclusion}(A) \not\subseteq \text{premise}(A) \wedge \text{premise}(A) \not\subseteq \cup_{j \neq i} CS_j^i \quad (8.3.4.I)$$

– si ag_i est **crédule** (noté $credulous(ag_i)$) alors :

$$want_concede(ag_i, H) \Leftrightarrow can_concede(ag_i, H) \quad (8.3.4.II)$$

On dit que $want_concede(ag_i, H)$ est la **tactique d'énonciation de la concession de l'hypothèse H par l'agent ag_i** . Elle dépend de l'attitude argumentative de l'agent locuteur : *crédule* ou *sceptique*.

Les autres actes de langage n'ont pas de tactique d'énonciation particulière.

Définition n° 8.3.4.VI [Tactiques d'énonciation des autres actes]

On note $want_question(ag_i, h)$, $want_unknow(ag_i, h)$, $want_challenge(ag_i, h)$ et $want_withdraw(ag_i, H)$, les prédicats définis de la manière suivante :

$$want_question(ag_i, H) \Leftrightarrow want_unknow(ag_i, H) \Leftrightarrow \\ want_challenge(ag_i, H) \Leftrightarrow want_withdraw(ag_i, H) \Leftrightarrow true \quad (8.3.4.III)$$

On dit que $want_question(ag_i, H)$, $want_unknow(ag_i, H)$, $want_challenge(ag_i, H)$ et $want_withdraw(ag_i, H)$, sont les **tactiques d'énonciation respectives d'une question, d'un aveu d'ignorance, d'une mise en doute et d'un abandon**.

L'exemple suivant permet d'illustrer ces définitions.

Exemple n° 8.3.4.II [Tactique d'énonciation] *L'agent ag_2 dispose d'arguments en faveur de chacun des candidats. Les arguments A' , A_1 et A_2 sont en faveur du candidat Jack. L'argument B est en faveur du candidat Jose. $\{A', A_1, A_2\}$ est l'ensemble des arguments acceptables selon l'agent ag_2 .*

Si l'agent ag_2 est confiant, les hypothèses $president(jose) \leftarrow$ et $president(jack) \leftarrow$ peuvent, toutes deux, être affirmées. Dans le cas contraire, seule l'hypothèse

president(jack) ← peut être affirmée.

Contrairement à l'argument A', les arguments B, A₁ et A₂ ne sont ni triviaux, ni issus du tableau d'engagement. C'est la raison pour laquelle, si l'agent ag₂ est crédule, alors les hypothèses president(jose) ← et president(jack) ← peuvent, toutes deux, être concédées. Dans le cas contraire, seule l'hypothèse president(jack) ← peut être concédée en vertu des arguments A₁ et A₂.

En résumé, la condition rationnelle d'énonciation d'un acte de langage détermine dans quelle mesure un agent peut émettre un message. Elles sont définies de la même manière, quel que soit l'agent envisagé. Ces conditions ne sont pas nécessairement mutuellement exclusives. C'est pourquoi nous avons défini les tactiques d'énonciation adoptées par les agents. Elles leur permettent de réaliser un choix individuel.

8.4 Synthèse

Nous avons présenté dans ce chapitre les éléments du modèle DIAL qui permettent aux agents d'une part d'échanger leurs connaissances et d'autre part de raisonner conjointement.

Pour échanger des connaissances, les agents doivent se comprendre. C'est la raison pour laquelle nous avons défini deux langages communs. D'une part, les agents partagent un même langage de représentation des connaissances. D'autre part, ils utilisent un même langage de communication d'agents.

Dans notre système multi-agents argumentatifs, les connaissances ainsi que les inclinaisons des agents diffèrent. Chaque agent est muni de son propre argumentaire et de sa propre échelle de valeur. Ils partagent toutefois des valeurs et des connaissances communes. Un agent argumentatif est un agent social dans la mesure où il dispose non seulement de ses croyances personnelles mais également d'un ensemble de tableaux d'engagement. Ces structures de données font le compte des déclarations perçues par l'agent et émises par ses interlocuteurs au cours des échanges.

Nous associons à chaque agent une théorie argumentative étendue. Cette dernière permet de prendre en considération les hypothèses émises par ses interlocuteurs. D'une part un agent interprète les hypothèses qui lui sont transmises. Ces hypothèses sont plus ou moins prises en considération en fonction de la valeur de réputation associée à l'agent qui a transmis cette information. D'autre part un agent émet des hypothèses.

Afin de l'interpréter, nous définissons la sémantique sociale d'un acte de langage. Selon l'acte reçu, les tableaux d'engagement sont mis à jour (ou non). Les hypothèses

ainsi adoptées doivent être évaluées. Leur priorité dépend de la compétence estimée de l'agent qui a transmis l'hypothèse.

Nous spécifions la sémantique argumentative d'un acte de langage d'une part en termes de conditions rationnelles d'énonciation et d'autre part en terme de tactiques d'énonciation. La condition rationnelle d'énonciation d'un acte de langage détermine dans quelle mesure un agent peut émettre un acte. Elles sont définies de la même manière, quel que soit l'agent envisagé. Ces conditions ne sont pas nécessairement mutuellement exclusives. L'agent concerné peut effectuer un choix individuel. La tactique d'énonciation adoptée par un agent dépend de son attitude.

Contrairement au modèle AMP [4, 46], le modèle DIAL permet aux agents de raisonner ensemble. Ils construisent leur argumentaire, non seulement à partir de leurs propres croyances mais également à partir des engagements pris par leurs interlocuteurs au cours des échanges. Les relations de priorité des agents étant statiques, la priorité d'une hypothèse adoptée au cours des échanges est évaluée sur la base de la compétence estimée de l'agent qui est à l'origine de l'information.

Grâce aux éléments du modèle DIAL que nous avons présentés ici, les agents échangent leurs connaissances et raisonnent conjointement. Mais dans quel but ? Les éléments du modèle DIAL présentés dans le chapitre suivant permettent de gérer ces flux d'information en régulant l'enchaînement des messages pour atteindre les buts de l'interaction. La notion de système dialectique multi-agents formalisée dans le chapitre suivant permet de relever ce défi.

“The debate is an institutionalized social tool to make society possible. The debate make it possible to have a political culture instead of living in a state of quarreling, repression and/or revolution where viewpoints are constantly in an eristic power struggle for dominance.”

Commitment in Dialogue by D. Walton & E. Krabbe

Chapitre 9

Systeme dialectique multi-agents

Sommaire

9.1	Introduction	155
9.2	Coup dialogique	156
9.3	Systeme dialectique	157
9.4	Protocole de dialogue	162
9.5	Règles de séquence	164
9.5.1	Réponse à une question	165
9.5.2	Réponse à une affirmation	166
9.5.3	Réponse à une mise en doute	170
9.5.4	Terminaison	170
9.6	Jeu de dialogue	171
9.6.1	Demande d'information	171
9.6.2	Persuasion	175
9.7	Synthèse	177

9.1 Introduction

Dans le chapitre précédent, nous avons présenté les éléments du modèle DIAL qui permettent aux agents d'échanger des informations et de raisonner conjointement. Lorsqu'un ensemble d'agents sociaux et autonomes interagissent directement, ils échangent des messages qui se répondent les uns aux autres dans le but de mener à bien l'interaction. Dans ce chapitre, nous présentons les éléments du modèle DIAL qui permettent de gérer le flux d'information en régulant l'enchaînement des messages pour atteindre les buts de l'interaction. La notion de système dialectique multi-agents formalisée dans ce chapitre permet de relever ce défi.

Dans un premier temps, nous allons définir les coups dialectiques, *i.e.* des messages munis d'attributs pour en contrôler la séquence (cf section 9.2). Nous présenterons ensuite, dans la section 9.3, le système dialectique multi-agents, *i.e.* le dispositif formel au travers duquel les agents jouent. Les dialogues sont des séquences de coups dialogiques conformes à un protocole. Nous définirons cette notion dans la section (cf section 9.4). Un protocole est constitué d'un ensemble de règles de séquence qui spécifie les réponses autorisées à un coup. Nous construisons à partir de ces règles deux protocoles de dialogue particuliers (cf section 9.6) : d'une part, un protocole de demande d'information et d'autre part un protocole de persuasion. Nous serons alors en mesure d'évaluer la qualité de tels dialogues.

9.2 Coup dialogique

Nous avons défini, dans la section 8.2.2, le langage de communication. La structure des messages ne permet pas de gérer leur enchaînement, or les actes de langage qui sont proférés au cours des échanges ne sont pas isolés. Ils se répondent les uns aux autres. C'est la raison pour laquelle nous définissons des coups dialogiques (*moves*). Les coups sont dotés de champs permettant de les lier les uns aux autres.

Les coups dialogiques proférés sont conformes à un **langage de coups** (noté $\mathcal{ML}_{\mathcal{U}}$). Chaque coup est muni d'un identifiant unique $move_k \in \mathcal{ML}_{\mathcal{U}}$.

Définition n° 9.2.0.VII [Coup dialogique]

Un **coup dialogique** $move_k \in \mathcal{ML}_{\mathcal{U}}$ est défini par un triplet

$move_k = \langle M_k, R_k, P_k \rangle$ où :

- $M_k = message(move_k)$ est un message conforme à la définition 8.2.2.I;
- $R_k = reply(move_k)$ est l'identifiant du coup auquel celui-ci répond ;
- $P_k = protocol(move_k)$ fait référence au protocole utilisé.

Un coup ($move_k$) est soit un coup d'initialisation ($reply(move_k) = \emptyset$) soit un coup de réponse ($reply(move_k) \neq \emptyset$). L'acte de langage d'un coup de réponse fait écho à l'un des actes précédemment émis. Les protocoles seront abordés dans la section 9.4.

En résumé, les coups dialogiques sont des structures qui permettent aux actes de langage de se répondre les uns aux autres. La section suivante porte sur le dispositif formel au travers duquel les coups dialogiques sont échangés.

9.3 Système dialectique

Un système dialectique est un dispositif formel au travers duquel deux agents jouent. Ils échangent des coups dialogiques pour aboutir à un accord. Un troisième agent, qui assiste au dialogue, arbitre en fonction des arguments qui sont mis à sa disposition et en fonction de son estimation de la compétence relative de chacun des joueurs.

Un système dialectique met en jeu trois agents. Deux joueurs dialoguent, en présence d'un témoin, à propos de la véracité d'une hypothèse, *i.e.* le thème. Les deux joueurs prennent la parole chacun à leur tour. L'échange est régulé par des règles dialectiques. La définition suivante décrit plus formellement un système dialectique multi-agents.

Définition n° 9.3.0.VIII [Système dialectique multi-agents]

Soit $\mathcal{AT}_{\Omega_A} = \langle \mathcal{T}_{\Omega_A}, V_{\Omega_A}, \text{promote}_{\Omega_A} \rangle$ une théorie argumentative multi-agents. Soit r_0 une règle issue du langage commun $\mathcal{L}_{\mathcal{U}}$. Un **système dialectique multi-agents** qui porte sur le **thème** r_0 est un septuplet

$DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ où :

- $N = \{\text{init}, \text{part}\}$ est un ensemble de deux agents argumentatifs appelés joueurs : un initiateur ($\text{init} \in \mathcal{U}_A$) et un partenaire ($\text{part} \in \mathcal{U}_A$) ;
- $\text{witness} \in \mathcal{U}_A$ est un troisième agent argumentatif, appelé témoin ;
- $\Omega_M \subseteq \mathcal{ML}_{\mathcal{U}}$ est un ensemble de coups bien formés ;
- H est l'ensemble des historiques, *i.e.* des séquences de coups bien formés $h_n = (\text{move}_1, \dots, \text{move}_n)$ définis tq $\forall 1 \leq k \leq n$:

1. le locuteur d'un coup est déterminé par une fonction de tour de parole :

$$\text{speaker}(\text{move}_k) = T(\text{move}_1, \dots, \text{move}_{k-1})$$

2. les coups doivent être conformes à une convention :

$$\text{move}_k \in \text{convention}(\text{move}_1, \dots, \text{move}_{k-1}).$$

- $T : H \rightarrow N$ est une fonction de tour de parole qui détermine, étant donné un historique, le locuteur du coup suivant. Si l'historique est de longueur nulle ou paire alors $T(h) = \text{init}$ sinon $T(h) = \text{part}$;
- $\text{convention} : H \rightarrow \Omega_M$ est une fonction qui, étant donné un historique, détermine l'ensemble des coups autorisés ;
- Z est l'ensemble des dialogues, *i.e.* les historiques de tailles maximales ;
- $u_{\text{init}}, u_{\text{part}} : H \rightarrow \{-1, 1\}$, *i.e.* les gains sont des fonctions qui déterminent le vainqueur d'un historique.

La définition du système dialectique multi-agents que nous proposons ici, s'inspire

de celle proposée par Henry Prakken [50, 49]. Toutefois ces deux définitions diffèrent quelque peu. Contrairement à la définition 4.3.2.II, nous ne faisons ici aucune hypothèse concernant la conviction initiale des joueurs sur le thème. C'est la raison pour laquelle on ne parle pas de valideur et de falsifieur mais d'initiateur et de partenaire. Un dialogue prend place entre le locuteur et l'allocutaire du coup d'initialisation. Ces deux **joueurs** jouent respectivement les rôles conventionnels suivants :

- l'**initiateur** (noté *init*), c'est-à-dire le locuteur du coup d'initialisation, correspond à l'agent qui débute le dialogue ;
- le **partenaire** (noté *part*), c'est-à-dire l'allocutaire du coup d'initialisation, correspond à l'agent qui est confronté à l'initiateur.

De cette manière, le système dialectique multi-agents que nous définissons ne se limite pas aux discussions critiques qui sont des dialogues de persuasion particuliers. Le système dialectique multi-agents permet de mettre en œuvre n'importe quel dialogue dont le but consiste à atteindre un accord.

Un joueur $p \in N$ est associé, pour un historique h , à la théorie argumentative étendue suivante :

$$\mathcal{AT}_p^*(h) = \langle \mathcal{T}_p^*(h), V_p^*, \text{promote}_p^*, \ll_p^* \rangle$$

Cette théorie argumentative étendue est, conformément à la définition 8.3.2.I, définie tq :

- $\mathcal{T}_p^*(h) = \mathcal{T}_p \cup \text{CS}_p^p(h)$ est la théorie personnelle étendue associée au joueur. Elle comprend d'une part la théorie personnelle de cet agent et d'autre part les engagements pris par son interlocuteur \bar{p} au cours de l'historique ;
- $V_p^* = V_p \cup \{v_p^p\}$ est l'ensemble étendu des valeurs personnelles associé au joueur. Cet ensemble est constitué d'une part de l'ensemble de ses valeurs personnelles et d'autre part de la valeur de réputation de son interlocuteur ;
- $\text{promote}_p^* : \mathcal{T}_p^* \rightarrow V_p^*$ est l'extension de la fonction promote_p . Elle associe à la théorie personnelle étendue un ensemble étendu de valeurs personnelles. D'une part, la théorie personnelle est associée aux valeurs personnelles. D'autre part, les engagements pris par son interlocuteur sont associés à la valeur de réputation de son interlocuteur ;
- \ll_p^* est la relation étendue de priorité du joueur.

$\mathcal{S}_p^*(h)$ dénote l'ensemble des arguments acceptables pour l'historique h par le joueur. Comme les joueurs débattent, chacun se considère plus compétent que son interlocuteur (cf définition 8.3.3.II). Pour la même raison, ils sont prudents (cf définition 8.3.4.IV) et sceptiques (cf définition 8.3.4.V).

Dans un système multi-agents argumentatifs, les agents qui ne participent pas directement au dialogue, sont les auditeurs des différents coups dialogiques sans en être les allocutaires. Ils sont appelés les **spectateurs** du dialogue (*bystander*). De

tels agents se livrent à une écoute flottante. Ils entendent les coups sans que leur locuteur ne s'attende à une réponse.

Le **témoin** est un des spectateurs du dialogue. Il ne dispose initialement d'aucune connaissance propre. En d'autres termes, sa théorie personnelle se limite à la théorie commune : $\mathcal{T}_{\text{witness}} = \mathcal{T}_{\Omega_A}$. On dit que le témoin est initialement ignorant. De même, l'ensemble de ses valeurs personnelles est réduit à l'ensemble des valeurs communes : $V_{\text{witness}} = V_{\Omega_A}$.

Le témoin est associé, pour un historique h , à la théorie argumentative étendue suivante :

$$\mathcal{AT}_{\text{witness}}^*(h) = \langle \mathcal{T}_{\text{witness}}^*(h), V_{\text{witness}}^*, \text{promote}_{\text{witness}}^*, \ll_{\text{witness}}^* \rangle$$

Cette théorie argumentative étendue est, conformément à la définition 8.3.2.I, définie tq :

- $\mathcal{T}_{\text{witness}}^*(h) = \mathcal{T}_{\text{witness}} \cup \text{CS}_{\text{init}}^{\text{witness}}(h) \cup \text{CS}_{\text{part}}^{\text{witness}}(h)$ est la théorie personnelle étendue associée au témoin. Elle comprend d'une part la théorie personnelle de cet agent et d'autre part les engagements pris par les deux joueurs au cours de l'historique ;
- $V_{\text{witness}}^* = V_{\Omega_A} \cup \{v_{\text{init}}^{\text{witness}}, v_{\text{part}}^{\text{witness}}\}$ est l'ensemble étendu des valeurs personnelles associé au témoin. Cet ensemble est constitué d'une part de l'ensemble de ses valeurs personnelles et d'autre part des valeurs de réputation des joueurs ;
- $\text{promote}_{\text{witness}}^* : \mathcal{T}_{\text{witness}}^* \rightarrow V_{\text{witness}}^*$ est l'extension de la fonction $\text{promote}_{\text{witness}}$. Elle associe à la théorie personnelle étendue un ensemble de valeurs personnelles. D'une part, la théorie commune est associée aux valeurs communes. D'autre part, un tableau d'engagement est associé à la valeur de réputation du débiteur ;
- \ll_{witness}^* est la relation étendue de priorité du témoin.

$\mathcal{S}_{\text{witness}}^*(h)$ dénote l'ensemble des arguments acceptables pour l'historique h par le témoin.

L'acceptabilité des arguments par le témoin dépend non seulement des arguments avancés mais également de son estimation de la compétence relative des joueurs. La relation de réputation étant une relation d'ordre total et strict, on distingue deux cas de figures. Soit le témoin considère l'initiateur plus compétent que le partenaire (part \prec_{witness} init). La valeur de réputation de l'initiateur est davantage prioritaire que la valeur de réputation du partenaire ($v_{\text{part}}^{\text{witness}} \ll_{\text{witness}}^* v_{\text{init}}^{\text{witness}}$). Soit le témoin considère l'initiateur moins compétent que le partenaire (init \prec_{witness} part). La valeur de réputation de l'initiateur est moins prioritaire que la valeur de réputation du partenaire ($v_{\text{init}}^{\text{witness}} \ll_{\text{witness}}^* v_{\text{part}}^{\text{witness}}$).

Le témoin peut (ou non) être convaincu par le thème. On dit que le thème est crédible si le témoin en est convaincu.

Définition n° 9.3.0.IX [Crédibilité]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur le thème r_0 . Soit $h \in H$ un historique. On dit que le **thème** r_0 est **crédible pour l'historique** h (noté $\text{provable}^h(r_0)$) ssi :

$$\exists A \in \mathcal{S}_{\text{witness}}^*(h) \text{ conclusion}(A) = r_0$$

Le témoin arbitre. Il est responsable du résultat. Les gains $(u_p)_{p \in N}$ déterminent si un joueur est un vainqueur sur un historique de la manière suivante :

1. Si $\text{provable}^h(r_0)$ alors $(u_p = 1 \Leftrightarrow \exists A \in \mathcal{S}_p^*(h_0) \text{ conclusion}(A) = r_0)$;
2. Si $\text{provable}^h(\neg r_0)$ alors $(u_p = 0 \Leftrightarrow \exists A \in \mathcal{S}_p^*(h_0) \text{ conclusion}(A) = r_0)$.

Contrairement au système dialectique présenté dans la section 4.3.2.1, le jeu n'est pas déterminé. En effet, nous ne faisons ici aucune hypothèse sur la conviction initiale des deux joueurs concernant le thème. Les joueurs sont déclarés vainqueur (ou non) sur la base de la théorie personnelle du témoin. La victoire d'un joueur dépend de la connaissance du témoin et de son échelle de valeurs sur les hypothèses échangées. La théorie argumentative étendue associée au témoin est constituée de l'ensemble des hypothèses notifiées dans les tableaux d'engagement $(\mathcal{CS}_{\text{init}}^{\text{witness}}(h) \cup \mathcal{CS}_{\text{part}}^{\text{witness}}(h))$ émises au cours de l'historique h . De plus, un dialogue débute sur un accord partiel. Les agents admettent conjointement la théorie commune (\mathcal{T}_{Ω_A}) . L'argumentaire dont dispose le témoin est dynamique. Il dépend de l'historique envisagé.

Il est probable que certains arguments n'aient pas été avancés au cours de l'historique envisagé. L'agent témoin ignore les arguments passés sous silence. On appelle **agent omniscient** (noté *omni*) un agent fictif qui disposerait de la connaissance de l'initiateur et de celle du partenaire. En d'autres termes, sa théorie personnelle est constituée de la théorie personnelle de l'initiateur et de la théorie personnelle du partenaire : $\mathcal{T}_{\text{omni}} = \mathcal{T}_{\text{init}} \cup \mathcal{T}_{\text{part}}$. Contrairement au témoin, l'agent omniscient dispose des arguments passés sous silence.

L'agent omniscient est muni de la théorie argumentative étendue suivante :

$$\mathcal{AT}_{\text{omni}} = \langle \mathcal{T}_{\text{omni}}, V_{\text{omni}}, \text{promote}_{\text{omni}}, \ll_{\text{omni}} \rangle$$

Cette théorie argumentative est, conformément à la définition 7.2.0.V, définie tq :

- $\mathcal{T}_{\text{omni}} = \mathcal{T}_{\text{init}} \cup \mathcal{T}_{\text{part}}$ est la théorie associée à l'agent omniscient. Elle comprend d'une part la théorie personnelle de l'initiateur et d'autre part la théorie personnelle du partenaire ;
- $V_{\text{omni}} = V_{\Omega_A} \cup V_{\text{init}}^{\text{omni}} \cup V_{\text{part}}^{\text{omni}}$ est l'ensemble des valeurs associé à l'agent omniscient. Cet ensemble est constitué de l'ensemble des valeurs communes, de l'ensemble des valeurs propres à l'initiateur et de l'ensemble des valeurs propres au partenaire ;

- $\text{promote}_{\text{omni}} : \mathcal{T}_{\text{omni}} \rightarrow V_{\text{omni}}$ est une fonction qui associe à chacune des règles une valeur. D'une part, les règles communes sont associées aux valeurs communes. D'autre part, les règles propres à l'initiateur sont associées aux valeurs propres à l'initiateur et les règles propres au partenaire sont associées aux valeurs propres au partenaire ;
- \ll_{omni} est la relation de priorité de l'agent omniscient.

$\mathcal{S}_{\text{omni}}$ dénote l'ensemble des arguments acceptables par l'agent omniscient.

L'acceptabilité des arguments par l'agent omniscient dépend de son estimation de la compétence relative des joueurs. Elle est identique à celle réalisée par le témoin. La relation de réputation étant une relation d'ordre total et strict, on distingue deux cas de figures. Soit le témoin considère l'initiateur plus compétent que le partenaire et les valeurs propres à l'initiateur sont plus prioritaires que les valeurs propres au partenaire :

$$\text{part} \prec_{\text{witness}} \text{init} \Rightarrow \forall v_{\text{part}} \in V_{\text{part}}^{\text{omni}} \forall v_{\text{init}} \in V_{\text{init}}^{\text{omni}} v_{\text{part}} \ll_{\text{omni}} v_{\text{init}}$$

Soit le témoin considère l'initiateur moins compétent que le partenaire et les valeurs propres à l'initiateur sont moins prioritaires que les valeurs propres au partenaire :

$$\text{init} \prec_{\text{witness}} \text{part} \Rightarrow \forall v_{\text{part}} \in V_{\text{part}}^{\text{omni}} \forall v_{\text{init}} \in V_{\text{init}}^{\text{omni}} v_{\text{init}} \ll_{\text{omni}} v_{\text{part}}$$

L'agent omniscient peut (ou non) est convaincu par le thème. On dit que le thème est un consensus si l'agent omniscient en est convaincu.

Définition n° 9.3.0.X [Consensus]

Soit $DS_{\Omega_M}(r_0, \mathcal{A}\mathcal{T}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur le thème r_0 . On dit que le thème r_0 est un **consensus** (notée $\text{consensus}(r_0)$) ssi :

$$\exists A \in \mathcal{S}_{\text{omni}} \text{ conclusion}(A) = r_0$$

Au terme d'un dialogue, le témoin ne dispose que d'une partie de l'argumentaire de chacun des joueurs. L'agent omniscient peut, quant à lui disposer d'un argument qui n'est ni à la disposition de l'initiateur ni à la disposition du partenaire. Plus formellement, les argumentaires des différents agents du système vérifient la propriété suivante :

$$\forall h \in Z \mathcal{A}(\mathcal{T}_{\text{witness}}^*(h)) \subseteq (\mathcal{A}(\mathcal{T}_{\text{init}}) \cup \mathcal{A}(\mathcal{T}_{\text{part}})) \subseteq (\mathcal{A}(\mathcal{T}_{\text{init}} \cup \mathcal{T}_{\text{part}}) = \mathcal{A}(\mathcal{T}_{\text{omni}}))$$

En conséquence, la conviction du témoin au terme du dialogue sur le thème n'est pas nécessairement la même que celle de l'agent omniscient. Il est probable que

certains arguments n'aient pas été avancés. Un dialogue est correct si le témoin est en mesure, au terme du dialogue, de se forger une conviction sur le thème. Un dialogue est complet si le témoin et l'agent omniscient ont la même conviction.

Définition n° 9.3.0.XI [Correction et complétude d'un dialogue]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur le thème r_0 . Soit $h \in Z$ un dialogue. Quitte à intervertir r_0 et $\neg r_0$,

– le dialogue est **correct** ssi :

$$\text{provable}^h(r_0)$$

– le dialogue est **complet** ssi :

$$\text{consensus}(r_0) \Rightarrow \text{provable}^h(r_0)$$

Le témoin est, au terme d'un dialogue correct, soit convaincu par le thème soit convaincu par sa négation. Un témoin a, au terme d'un dialogue complet, la même conviction que l'agent omniscient sur le thème.

Un argument de l'agent omniscient qui n'est dans l'argumentaire d'aucun des joueurs n'est pas non plus dans l'argumentaire du témoin au terme du dialogue. La conviction du témoin au terme du dialogue et celle de l'agent omniscient peuvent diverger. Le dialogue envisagé dans la section E.3 illustre ce cas de figure. La propriété suivante est une condition nécessaire mais pas suffisante pour garantir la complétude des dialogues :

$$(\mathcal{A}(\mathcal{T}_{\text{init}}) \cup \mathcal{A}(\mathcal{T}_{\text{part}}) = \mathcal{A}(\mathcal{T}_{\text{init}} \cup \mathcal{T}_{\text{part}})) \quad (9.3.0.I)$$

Si cette propriété n'est pas vérifiée, un dialogue ne peut pas être complet.

En résumé, un dialogue prend place entre deux joueurs, l'initiateur et le partenaire. Ces deux joueurs échangent des coups dont le témoin est un auditeur. Il est probable que certains arguments n'aient pas été avancés au cours du dialogue. Le témoin ne dispose pas de toute la connaissance du système dialectique au terme du dialogue. La qualité d'un dialogue dépend du protocole utilisé. Nous définissons cette notion dans la section suivante.

9.4 Protocole de dialogue

Un protocole de dialogue est constitué d'un ensemble de règles dialectiques qui spécifient les coups autorisés.

Un historique est un dialogue en cours de réalisation. Il est constitué d'une séquence finie de coup bien formés. À l'exception du coup d'initialisation, un coup répond à l'un des coups précédents dans l'historique. Les retours arrières (*backtracks*) sont possibles. La ligne persuasive d'un historique est la sous-séquence dans laquelle ces retours arrières ont été évincés.

Définition n° 9.4.0.XII [Ligne persuasive]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur le thème r_0 . Soit $h_k = (\text{move}_1, \dots, \text{move}_k) \in H$ un historique. La **ligne persuasive** de h_k (notée $\text{line}(h_k)$) est la plus petite sous-séquence de h_k tq :

$$\begin{aligned} & \text{move}_1, \text{move}_k \in \text{line}(h_k) \\ \forall \text{move}_i \in \text{line}(h_k) & \text{move}_i \in \text{convention}(\text{move}_1, \dots, \text{move}_{i-1}) \end{aligned}$$

En d'autres termes, les coups de réponse dans une ligne persuasive font référence au coup précédent.

Les coups doivent, pour être bien formés, vérifier les propriétés suivantes :

Définition n° 9.4.0.XIII [Coups bien formés]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur le thème r_0 . Soit $h_{k-1} = (\text{move}_1, \dots, \text{move}_{k-1}) \in H$. Un coup move_k est bien formé ssi :

1. si $h_{k-1} = \emptyset$ alors $\text{act}(\text{move}_k) = \text{question}(r_0)$, $\text{speaker}(\text{move}_k) = \text{init}$ et $\text{hearer}(\text{move}_k) = \text{part}$;
2. sinon $\exists \text{move}_i \in h_{k-1}$ tq :
 - (a) $\text{reply}(\text{move}_k) = \text{move}_i$ avec $\text{speaker}(\text{move}_k) \neq \text{speaker}(\text{move}_i)$;
 - (b) si $\text{move}_i \in \text{line}(h_{k-1})$ et $\text{act}(\text{move}_i) = \text{assert}(H)$ alors $\text{act}(\text{move}_k) \neq \text{assert}(H')$ avec $H \cap H' \neq \emptyset$;
 - (c) $\text{protocol}(\text{move}_k) = \text{protocol}(\text{move}_1)$.

Un coup d'initialisation est bien formé si c'est une question sur le thème, proférée par l'initiateur, à l'intention du partenaire (cf condition 1). Un coup de réponse émis par un joueur est bien formé s'il répond à un des coups précédents émis par l'autre joueur (cf condition 2a). Afin d'éviter la présence de boucles dans un dialogue, la redondance d'hypothèses est interdite dans les affirmations d'une même ligne persuasive (cf condition 2b). Bien évidemment, le même protocole doit être conservé au cours des échanges (cf condition 2c).

Un **protocole de dialogue** (que nous appelons ici simplement protocole) est le nom qui identifie l'ensemble des règles dialectiques. Celles-ci spécifient les coups autorisés (ou non) et sont compilées dans la fonction convention.

Nous n'envisagerons dans la suite de ce manuscrit que des protocoles à coups uniques. En d'autres termes, le joueur ne peut émettre qu'un seul coup par tour de parole. Toutefois, on distingue les deux cas de figure suivants. Un **protocole** peut être soit à réponses uniques soit à réponses multiples.

Définition n° 9.4.0.XIV [Protocole (à réponses uniques/multiples)]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur le thème r_0 . Le protocole utilisé est :

– soit à **réponses uniques** ssi

$$\forall h_{k-1} \in H \forall \text{move}_k \in \text{convention}(h_{k-1}), \text{reply}(\text{move}_k) = \text{move}_{k-1};$$

– soit à **réponses multiples** ssi

$$\forall h_{k-1} \in H \forall \text{move}_k \in \text{convention}(h_{k-1}) \exists \text{move}_i \in h_{k-1}, \text{reply}(\text{move}_k) = \text{move}_i.$$

Dans le premier cas, un joueur ne peut apporter qu'une seule réponse à chacun des coups de son adversaire. Les historiques sont tous des lignes persuasives. Dans le second cas, un joueur peut essayer plusieurs actes de langage pour répondre à ceux de son adversaire.

Les règles de séquence présentées dans la section suivante permettent de décrire des protocoles à réponses uniques.

9.5 Règles de séquence

Les règles de séquence précisent quelles sont les réponses autorisées à un coup. Comme les conditions rationnelles d'énonciation (cf section 8.3.4.1), les règles de séquence sont des règles communes à tous les agents du système. Ces derniers ont non seulement leurs propres tactiques argumentatives (cf section 8.3.4.2) mais également leurs propres tactiques conventionnelles qui leur permettent de sélectionner la réponse privilégiée à un coup.

Considérons un système dialectique multi-agents qui porte sur le thème r_0 ($DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$) et notamment l'un de ses historiques $h_{k-1} = (\text{move}_1, \dots, \text{move}_{k-1}) \in H$. Une **règle de séquence** (*sequence rules*) spécifie quels actes de langage peuvent répondre à l'un des actes de langage précédemment émis par l'interlocuteur. On distingue deux cas de figure. Soit le second acte résiste face au premier, soit le second acte abdique face au premier.

Un dialogue débute toujours par une question sur le thème.

9.5.1 Réponse à une question

D'après la règle de « Question/Réponse », l'allocutaire d'une question ($\text{question}(h)$) peut répondre : soit par une **confirmation** ($\text{assert}(h)$), soit par une **infirmation** ($\text{assert}(\neg h)$), soit par un **aveu d'ignorance** ($\text{unknow}(h)$).

Cette règle est représentée dans la table 9.5.1. Donner son opinion, une confirmation ou une infirmation, est un acte de résistance alors que l'aveu d'ignorance est un acte d'abdication.

Règle de séquence	Acte de langage	Actes résistants	Actes abdiquants
$\text{sr}_{Q/A}$	$\text{question}(h)$	$\text{assert}(h)$ $\text{assert}(\neg h)$	$\text{unknow}(h)$

Tab. 9.5.1.I: Règle de « Question/Réponse » : question et réponses potentielles.

Les tactiques d'énonciation de ces différents actes, qu'elles résistent ou abdiquent face à une question, ne sont pas nécessairement mutuellement exclusives. Un agent n'est pas en mesure, avec cette seule règle, d'émettre une réponse unique. Les deux premières tactiques conventionnelles spécifient la réponse privilégiée à une question.

Tactiques de réponse à une question. Suite à une question, l'agent ag_i qui peut donner son opinion est **coopératif** (noté $\text{benevolent}(\text{ag}_i)$) s'il accède à la requête. Dans le cas contraire, il est **égoïste** (noté $\text{egoist}(\text{ag}_i)$). Étant donné que les joueurs débattent, ils sont coopératifs.

La tactique d'énonciation d'une confirmation et celle d'une infirmation ne sont pas mutuellement exclusives. Dans ce cas de figure, l'agent doit trancher parmi ces deux réponses. C'est l'objet de la seconde tactique conventionnelle.

Suite à une question, un agent ag_i confiant peut répondre indifféremment par une confirmation ou une infirmation. Il est **positif** (noté $\text{positive}(\text{ag}_i)$) s'il confirme ou **néгатif** s'il infirme (noté $\text{negative}(\text{ag}_i)$).

L'algorithme 9.5.1 décrit le processus de sélection de l'acte de langage qui répond à une question.

La réponse préconisée par cet algorithme est unique et en conformité avec la règle de « Question/Réponse ». En d'autres termes, un agent peut toujours répondre de manière univoque par la règle de « Question/Réponse ».

L'annexe E met à disposition deux exemples de réponse à une question : dans la section E.2 et dans la section E.4.

Comme cela est prescrit par la règle de « Question/Réponse », le dialogue peut se poursuivre par une affirmation.

```

Entrée : question( $h$ ),  $ag_i$ 
Sortie : act(move $_k$ )
si egoist( $ag_i$ ) alors
  | // want_unknow( $ag_i, h$ );
  | unknow( $h$ );
sinon
  | // benevolent( $ag_i$ ) ;
  | si  $\neg$  want_assert( $ag_i, h$ )  $\wedge$   $\neg$  want_assert( $ag_i, \neg h$ ) alors unknow( $h$ )
  | si want_assert( $ag_i, h$ )  $\wedge$   $\neg$  want_assert( $ag_i, \neg h$ ) alors assert( $h$ )
  | si  $\neg$  want_assert( $ag_i, h$ )  $\wedge$  want_assert( $ag_i, \neg h$ ) alors assert( $\neg h$ );
  | si want_assert( $ag_i, h$ )  $\wedge$  want_assert( $ag_i, \neg h$ ) alors
    | si positive( $ag_i$ ) alors
      | | assert( $h$ )
      | sinon
      | | #negative( $ag_i$ );
      | | assert( $\neg h$ );
      | fin
    | fin
  | fin
fin

```

Fig. 9.5.1.I: Réponse à une question

9.5.2 Réponse à une affirmation

D'après la règle d' « Affirmation/Accueil », l'allocutaire d'une affirmation (assert(H)) peut répondre : soit par un **accueil favorable** (concede(H)), soit par un **accueil défavorable** (refuse(h) avec $h \in H$), soit par une **mise en doute** (challenge(h) avec $h \in H$).

Cette règle est représentée dans la table 9.5.2. La concession et l'accueil défavorable sont des actes d'abdication alors qu'une mise en doute est un acte de résistance.

Règle de séquence	Acte de langage	Actes résistants	Actes abdiquants
sr _{A/W}	assert(H)	challenge(h), $h \in H$	concede(H) refuse(h), $h \in H$

Tab. 9.5.2.I: Règle d' « Affirmation/Accueil » : affirmation et réponses potentielles.

Contrairement à la règle d' « Affirmation/Accueil », la règle d' « Affirmation/Réfutation » ne permet pas de répondre à une affirmation par un accueil défavorable, qui équivaut à concede($\neg h$), mais par une réfutation, qui équivaut à assert($\neg h$). Ces actes de

langage différent par leurs conditions d'énonciation (cf définition 8.3.4.I et définition 8.3.4.II). De plus, ces deux réponses se distinguent par leurs réponses possibles. Comme nous le verrons dans la section 9.5.4, un accueil défavorable, contrairement à une réfutation, met un terme à la ligne persuasive.

Cette règle est représentée dans la table 9.5.2. Concéder est un acte d'abdication alors qu'une mise en doute ou une réfutation est un acte de résistance.

Règle de séquence	Acte de langage	Actes résistants	Actes abdiquants
$sr_{A/R}$	$assert(H)$	$challenge(h), h \in H$ $reject(h), h \in H$	$concede(H)$

Tab. 9.5.2.II: Règle d' « Affirmation/Réfutation » : affirmation et réponses potentielles.

Tactiques de réponse à une affirmation. Quatre autres stratégies conventionnelles, associées aux deux règles de séquence précédentes, permettent de sélectionner la réponse privilégiée.

Lors d'une séquence d' « Affirmation/Accueil », l'agent ag_i qui peut, soit mettre en doute, soit donner son opinion est : **argumentatif** (noté $argumentative_{A/W}(ag_j)$) s'il met en doute, **ouvert d'esprit** (noté $open_mind_{A/W}(ag_j)$) s'il donne son opinion.

La tactique lors d'une séquence d' « Affirmation/Réfutation » est définie de manière similaire. On note $argumentative_{A/R}(ag_j)$ et $open_mind_{A/R}(ag_j)$ les deux attitudes possibles.

Comme les joueurs débattent, ils sont ouverts d'esprit.

Lors d'une séquence d' « Affirmation/Accueil », un agent ag_i confiant ou crédule qui peut indifféremment répondre par un accueil favorable ou par un accueil défavorable est : **optimiste** (noté $optimistic_{A/R}(ag_i)$) s'il concède ; **pessimiste** (noté $pessimistic_{A/W}(ag_i)$) s'il refuse.

La tactique lors d'une séquence d' « Affirmation/Réfutation » est définie de manière similaire. Les attitudes correspondantes sont notées $optimistic_{A/R}(ag_i)$ et $pessimistic_{A/R}(ag_i)$.

L'algorithme 9.5.2 décrit le processus de sélection de l'acte de langage en conformité avec la règle d' « Affirmation/Accueil ».

```

Entrée : assert( $H$ ),  $ag_i$ 
Sortie : act(move $_k$ )
si argumentative $_{A/W}(ag_i)$  alors
  | // want_challenge( $ag_i, h$ );
  | select( $h, H$ );
  | challenge( $h$ );
sinon
  | // open_mind $_{A/W}(ag_i)$  ;
  | si  $\neg$  want_concede( $ag_i, H$ )  $\wedge$   $\nexists h \in H$  want_concede( $ag_i, \neg h$ ) alors
  | | select( $h, H$ );
  | | challenge( $h$ );
  | fin
  | si want_concede( $ag_i, H$ )  $\wedge$   $\nexists h \in H$  want_concede( $ag_i, \neg h$ ) alors
  | | concede( $H$ );
  | fin
  | si  $\neg$  want_concede( $ag_i, H$ )  $\wedge$   $\exists h \in H$  want_concede( $ag_i, \neg h$ ) alors
  | | refuse( $h$ );
  | fin
  | si want_concede( $ag_i, H$ )  $\wedge$   $\exists h \in H$  want_concede( $ag_i, \neg h$ ) alors
  | | si optimistic $_{A/W}(ag_i)$  alors
  | | | concede( $H$ )
  | | | sinon
  | | | //pessimistic $_{A/W}(ag_i)$ ;
  | | | refuse( $h$ );
  | | | fin
  | | fin
  | fin
fin

```

Fig. 9.5.2.I: Réponse à une affirmation selon la règle d' « Affirmation/Accueil »

L'algorithme 9.5.2 décrit le processus de sélection de l'acte de langage en conformité avec la règle de « Affirmation/Réfutation ». La fonction $select(h, H)$ permet de sélectionner l'une des hypothèses h de H .

```

Entrée : assert( $H$ ),  $ag_i$ 
Sortie : act(move $_k$ )
si argumentative $_{A/R}(ag_i)$  alors
| // want_challenge( $ag_i, h$ );
| select( $h, H$ );
| challenge( $h$ );
sinon
| // open_mind $_{A/R}(ag_i)$  ;
| si  $\neg$  want_concede( $ag_i, H$ )  $\wedge$   $\nexists h \in H$  want_assert( $ag_i, \neg h$ ) alors
| | select( $h, H$ );
| | challenge( $h$ );
| | fin
| | si want_concede( $ag_i, H$ )  $\wedge$   $\nexists h \in H$  want_assert( $ag_i, \neg h$ ) alors
| | | concede( $H$ );
| | | fin
| | si  $\neg$  want_concede( $ag_i, H$ )  $\wedge$   $\exists h \in H$  want_assert( $ag_i, \neg h$ ) alors
| | | reject( $h$ ) ;
| | | fin
| | si want_concede( $ag_i, H$ )  $\wedge$   $\exists h \in H$  want_assert( $ag_i, \neg h$ ) alors
| | | si optimistic $_{A/R}(ag_i)$  alors
| | | | concede( $H$ )
| | | | sinon
| | | | //pessimistic $_{A/R}(ag_i)$ ;
| | | | reject( $h$ )
| | | | fin
| | | fin
| | fin
| fin
fin

```

Fig. 9.5.2.II: Réponse à une affirmation selon la règle d' « Affirmation/Réfutation »

La réponse préconisée par chacun de ces algorithmes est unique et en conformité avec la règle correspondante. En d'autres termes, un agent peut toujours répondre de manière univoque par chacune de ces deux règles. L'annexe E met à disposition cinq exemples de réponse à une affirmation. Les deux premières réponses utilisent la règle d' « Affirmation/Accueil » (cf section E.2 et section E.2). Les trois suivantes utilisent la règle d' « Affirmation/Réfutation » (cf section E.4, section E.4 et section E.4).

Comme cela est prescrit par les deux règles précédentes, le dialogue peut se pour-

suivre par une mise en doute.

9.5.3 Réponse à une mise en doute

D'après la règle de « Mise en doute/Réponse », un agent ag_i peut répondre à une mise en doute ($challenge(h)$) par une **justification** ($assert(H)$, avec $H \vdash_{\mathcal{U}} h$), ou par un **abandon** $withdraw(h)$.

Cette règle est représentée dans la table 9.5.3. Abandonner est un acte d'abdication alors qu'une justification est un acte de résistance.

Règle de séquence	Acte de langage	Actes résistants	Actes abdiquants
$sr_{C/A}$	$challenge(h)$	$assert(H)$, avec $H \vdash_{\mathcal{U}} h$	$withdraw(h)$

Tab. 9.5.3.I: Règle de « Mise en doute/Réponse » : mise en doute et réponses potentielles.

Suite à une mise en doute, un agent répond dans la mesure du possible par un argument. La réponse préconisée par cette tactique est unique et en conformité avec la règle de « Mise en doute/Réponse ». En d'autres termes, un agent peut toujours répondre de manière univoque par la règle de « Mise en doute/Réponse ». L'annexe E met à disposition quatre exemples de réponse à une mise en doute (cf section E.2, section E.2, section E.4 et section E.4).

9.5.4 Terminaison

D'après la règle de terminaison, les actes de langage qui n'ont pas encore été envisagés, mettent fin à la ligne persuasive.

Cette règle est représentée dans la table 9.5.4. L'ensemble des actes d'abdication dans les règles précédentes mettent un terme à la ligne persuasive.

Règle de séquence	Acte de langage	Actes résistants	Actes abdiquants
sr_T	$unknow(H)$	\emptyset	\emptyset
	$concede(H)$	\emptyset	\emptyset
	$withdraw(H)$	\emptyset	\emptyset

Tab. 9.5.4.I: Règle de terminaison.

L'annexe E met à disposition deux exemples de terminaison (cf section E.4 et section E.4).

Les règles de séquence précédemment énumérées permettent de mettre en exergue les deux rôles dialectiques suivants :

1. le locuteur d'une mise en doute joue le rôle de **challengeur** ;
2. le locuteur d'une affirmation, que cela soit une confirmation, une infirmation, une réfutation ou une justification, joue le rôle d'**argumentateur**.

Ces rôles peuvent évoluer (ou non) au cours d'un historique selon le protocole envisagé.

9.6 Jeu de dialogue

Dans la section 3.3, nous avons présenté la catégorisation des dialogues proposée par Walton et Krabbe [67]. En identifiant la situation initiale ainsi que les buts des dialogues, cette classification distingue six catégories primaires de dialogue : la persuasion, la négociation, l'enquête, la délibération, la demande d'information et l'éristique. Afin d'illustrer le cadre formel proposé précédemment, nous présenterons ici deux protocoles de dialogue particuliers : un protocole de demande d'information à réponses uniques et un protocole de persuasion à réponses uniques.

9.6.1 Demande d'information

Une demande d'information intervient lorsque l'initiateur souhaite disposer d'une connaissance qu'il suppose être détenue par le partenaire. La situation initiale est donc asymétrique. Le partenaire, contrairement à l'initiateur, a une conviction sur le thème. Le but du dialogue, c'est-à-dire le but partagé par les agents, consiste à diffuser une connaissance. Nous proposons ici un protocole de dialogue particulier. Les dialogues conformes à ce protocole sont des demandes d'information.

Dans un premier temps, nous allons démontrer la terminaison des demandes d'information, quelle que soit la situation initiale. Dans un second temps, nous serons en mesure de démontrer qu'une demande d'information est correcte et complète sous certaines conditions concernant la situation initiale des joueurs.

Le protocole de dialogue proposé ici est un protocole à réponses uniques.

Définition n° 9.6.1.I [Protocole de demande d'information]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur le thème r_0 . Le protocole est appelé **protocole de demande d'information à réponses uniques** (noté ISP) ssi :

1. c'est un protocole à réponses uniques (cf définition 9.4.0.XIV) ;
2. la fonction convention vérifie les quatre règles de séquence suivantes :
 - (a) $sr_{Q/A}$, i.e. la règle de « Question/Réponse » (cf table 9.5.1) ;
 - (b) $sr_{A/W}$, i.e. la règle d' « Affirmation/Accueil » (cf table 9.5.2) ;
 - (c) $sr_{C/A}$, i.e. la règle de « Mise en doute/Réponse » (cf table 9.5.3) ;
 - (d) sr_T , i.e. la règle de terminaison (cf table 9.5.4).

La convention est constituée de la règle de « Question/Réponse », de la règle d' « Affirmation/Accueil », de la règle de « Mise en doute/Réponse » et de la règle de terminaison. Ces règles de séquence sont représentées dans la table 9.6.1.

Règles de séquence	Actes de langage	Actes résistants	Actes abdiquants
$sr_{Q/A}$	question(h)	assert(h) assert($-h$)	unknow(h)
$sr_{A/W}$	assert(H)	challenge(h), $h \in H$	concede(H) refuse(h), $h \in H$
$sr_{C/A}$	challenge(h)	assert(H), avec $H \vdash_{\cup} h$	withdraw(h)
sr_T	unknow(H)	\emptyset	\emptyset
	concede(H)	\emptyset	\emptyset
	withdraw(H)	\emptyset	\emptyset

Tab. 9.6.1.I: Règles de séquence du protocole de demande d'information à réponses uniques.

La figure 9.6.1.I représente un jeu de demande d'information sous forme normale extensive où les arêtes sont associées à des coups dialogiques et les nœuds sont des **situations de jeu**. Leur exposant indique lequel des joueurs joue le coup suivant. Par exemple, le partenaire est le locuteur du coup suivant dans la situation de jeu 3.3^{part}. Les situations 2.1, 3.1, 3.2, 4.1, 5.1, 5.2, 6.1... sont des situations de fin de jeu.

Un agent est à l'initiative d'une demande d'information lorsqu'il souhaite disposer de l'avis du partenaire sur le thème. Si le partenaire ne dispose d'aucun argument en faveur ou en défaveur du thème, celui-ci répond par un aveu d'ignorance qui met un terme au dialogue (cf situation de jeu 2.1). Dans le cas contraire, le but du dialogue,

c'est-à-dire le but partagé par les deux joueurs consiste à diffuser cette connaissance. Le témoin doit au terme du dialogue disposer de cette connaissance.

La terminaison des demandes d'information peut être garantie, quelle que soit la situation initiale des joueurs.

Théorème n° 9.6.1.I [Terminaison d'une demande d'information]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur le thème r_0 et qui emploie un protocole de demande d'information à réponses uniques. La demande d'information ($h \in Z$) est finie.

L'existence et l'unicité de la réponse à un coup, et ce quelle que soit la règle de séquence envisagée, ont été argumentées dans la section 9.5. Ses propriétés sont nécessaires pour la démonstration de ce théorème qui se trouve dans la section D.2.

La terminaison des demandes d'information est garantie dans tous les cas de figure. A l'inverse, certaines hypothèses sur la situation initiale des joueurs sont nécessaires pour garantir la correction des demandes d'information.

D'après le théorème suivant, une demande d'information est correcte si l'initiateur ne dispose initialement d'arguments ni en faveur ni en défaveur du thème et si le partenaire est convaincu par le thème.

Théorème n° 9.6.1.II [Correction d'une demande d'information]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur le thème r_0 et qui emploie un protocole de demande d'information à réponses uniques. Si la situation initiale est telle que :

– l'initiateur ne dispose initialement d'aucun argument en faveur ou en défaveur de r_0 :

$$\forall A \in \mathcal{A}(\mathcal{T}_{\text{init}}) \text{ conclusion}(A) \neq r_0 \wedge \text{conclusion}(A) \neq \neg r_0$$

– le partenaire est initialement convaincu de r_0 :

$$\exists A \in \mathcal{S}_{\text{part}} \text{ conclusion}(A) = r_0$$

alors la demande d'information ($h \in Z$) est correcte.

Afin de simplifier la lecture de ce théorème, nous supposons le partenaire convaincu de r_0 . Nous aurions pu tout aussi bien considérer le partenaire convaincu du contraire. Il suffit de remplacer dans ce théorème r_0 par $\neg r_0$. La démonstration de ce théorème se trouve dans la section D.3.

Le témoin est, au terme d'une telle demande d'information, soit convaincu par le thème soit convaincu du contraire. Sa conviction peut être différente de celle de l'agent omniscient. Pour que la conviction de ces deux agents soient identiques, les conditions sur la situation initiale doivent être plus restrictives.

D'après le théorème suivant, une demande d'information est complète si l'initiateur est initialement ignorant et si le partenaire est convaincu par le thème.

Théorème n° 9.6.1.III [Complétude d'une demande d'information]

Soit $DS_{\Omega_M}(r_0, AS_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur le thème r_0 et qui emploie un protocole de demande d'information. Si la situation initiale est telle que :

– l'initiateur est initialement ignorant :

$$\mathcal{T}_{init} = \mathcal{T}_{\Omega_A}$$

– le partenaire est initialement convaincu de r_0 :

$$\exists A \in \mathcal{S}_{part}^* \text{conclusion}(A) = r_0 \quad (9.6.1.I)$$

La demande d'information ($h \in Z$) est complète.

Pour les mêmes raisons évoquées précédemment, nous supposons le partenaire convaincu de r_0 . La démonstration de ce théorème se trouve dans la section D.4.

Afin d'illustrer les propriétés des demandes d'information, nous proposons dans l'annexe E deux exemples de demande d'information. La première demande d'information est correcte et complète (cf section E.2). La seconde demande d'information est correcte mais incomplète (cf section E.3).

Une demande d'information est incomplète si l'agent omniscient dispose d'un argument qui n'est initialement ni à disposition de l'initiateur ni à disposition du partenaire (cf équation 9.3.0.I). En d'autres termes, si l'initiateur n'est pas initialement ignorant alors l'argumentaire de l'agent omniscient peut contenir un argument qui n'est à la disposition d'aucun joueur. Cet argument sera, par conséquence, passé sous silence.

Par construction du jeu de demande d'information, les mises en doute sont toujours émises par l'initiateur. Il joue le rôle de **challengeur** tout au long du dialogue. De même, les justifications sont toujours énoncées par le partenaire. Il joue le rôle d'**argumentateur** tout au long du dialogue. Ce dernier a la charge de la preuve. Comme nous le verrons dans la section suivante, ces rôles dialectiques peuvent évoluer au cours d'une persuasion.

9.6.2 Persuasion

Un dialogue de persuasion intervient lorsque l'initiateur souhaite confronter son avis avec celui du partenaire. Dans la situation de jeu initiale, les joueurs sont en désaccord sur le thème. Le but du dialogue consiste à résoudre verbalement le conflit. Le témoin doit, au terme du dialogue, avoir une conviction. Nous proposons ici un protocole de dialogue particulier. Les dialogues conformes à ce protocole sont des persuasions.

Dans un premier temps, nous allons démontrer la terminaison des persuasions, quelle que soit la situation initiale. Dans un second temps, nous serons en mesure de démontrer qu'une persuasion est correcte sous certaines conditions concernant la situation initiale des joueurs et leurs tactiques de jeu.

Le protocole utilisé est un protocole à réponses uniques.

Définition n° 9.6.2.I [Protocole de persuasion]

Soit $DS_{\Omega_M}(r_0, AS_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur le thème r_0 . Le protocole est appelé **protocole de persuasion à réponses uniques** (noté *PP*) ssi ;

1. il est conforme à la définition 9.3.0.X ;
2. c'est un protocole à réponses uniques (cf définition 9.4.0.XIV) ;
3. la fonction convention vérifie les quatre règles de séquence suivantes :
 - (a) $sr_{Q/A}$, i.e. la règle de « Question/Réponse » (cf table 9.5.1) ;
 - (b) $sr_{A/R}$, i.e. la règle d' « Affirmation/Réfutation » (cf table 9.5.2) ;
 - (c) $sr_{C/A}$, i.e. la règle de « Mise en doute/Réponse » (cf table 9.5.3) ;
 - (d) sr_T , i.e. la règle de terminaison (cf table 9.5.4).

La convention se distingue de celle d'un protocole de demande d'information par la règle d' « Affirmation/Réfutation » qui est utilisée en lieu et place de la règle d' « Affirmation/Accueil ». Ces règles de séquence sont représentées dans la table 9.6.2.

La figure 9.6.2.I représente un jeu de persuasion sous forme normale extensive. Contrairement à la figure 9.6.1.I, la situation de jeu 3.1^{part} n'est pas une situation de fin de jeu. L'initiateur ne peut pas mettre fin au dialogue en exprimant son désaccord. Ainsi, le partenaire n'a pas à lui seul la charge de la preuve. Par construction du jeu de persuasion, les joueurs ont la possibilité de changer de rôle dialectique au cours du dialogue.

Règles de séquence	Actes de langage	Actes résistants	Actes abdiquants
$sr_{Q/A}$	$question(h)$	$assert(h)$ $assert(\neg h)$	$unknow(h)$
$sr_{A/R}$	$assert(H)$	$challenge(h), h \in H$ $reject(h), h \in H$	$concede(H)$
$sr_{C/A}$	$challenge(h)$	$assert(H), \text{ avec } H \vdash_{\mathcal{U}} h$	$withdraw(h)$
sr_T	$unknow(H)$	\emptyset	\emptyset
	$concede(H)$	\emptyset	\emptyset
	$withdraw(H)$	\emptyset	\emptyset

Tab. 9.6.2.I: Règles de séquence du protocole de persuasion à réponses uniques.

Un agent initie une persuasion lorsqu'il souhaite convaincre ses interlocuteurs de la véracité (ou non) d'une hypothèse. Si le partenaire dispose d'une conviction contraire alors le but du dialogue, c'est-à-dire le but partagé par les deux joueurs, consiste à convaincre le témoin.

La terminaison des persuasions peut être garantie, quelle que soit la situation initiale des joueurs.

Théorème n° 9.6.2.I [Terminaison d'une persuasion]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, witness, H, T, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur le thème r_0 et qui emploie un protocole de persuasion à réponses uniques. La persuasion ($h \in Z$) est finie.

L'existence et l'unicité de la réponse à un coup, et ce quelle que soit la règle de séquence envisagée, ont été argumentées dans la section 9.5. Ses propriétés sont nécessaires pour la démonstration de ce théorème qui se trouve dans la section D.5.

Nous avons démontré que la terminaison des persuasions est garantie dans tous les cas de figure. A l'inverse, certaines hypothèses sur la situation initiale des joueurs sont nécessaires pour garantir la correction d'une persuasion.

D'après le théorème suivant, une persuasion est correcte si l'initiateur et le partenaire ont des convictions différentes sur le thème.

Théorème n° 9.6.2.II [Correction d'une persuasion]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur le thème r_0 et qui emploie un protocole de persuasion à réponses uniques.

Si la situation initiale des joueurs est telle que :

– l'initiateur est convaincu de $\neg r_0$:

$$\exists A \in \mathcal{S}_{\text{init}}^* \text{conclusion}(A) = \neg r_0$$

– le partenaire est convaincu de r_0 :

$$\exists A \in \mathcal{S}_{\text{part}}^* \text{conclusion}(A) = r_0$$

alors la persuasion ($h \in Z$) est correcte.

Afin de simplifier la lecture de ce théorème, nous supposons l'initiateur convaincu de $\neg r_0$ et le partenaire convaincu de r_0 . Nous aurions pu tout aussi bien considérer l'inverse. Il suffit de remplacer dans ce théorème r_0 par $\neg r_0$, et réciproquement. La démonstration de ce théorème se trouve dans la section D.6.

L'initiateur dispose initialement d'une conviction sur le thème. Il n'est pas ignorant. En conséquence, l'équation 9.3.0.I n'est pas forcément vérifiée. La complétude d'une persuasion ne peut pas être garantie. Nous proposons dans la section E.4 un exemple de persuasion. Conformément au théorème 9.6.2.II, cette persuasion est correcte.

9.7 Synthèse

Nous avons présenté dans ce chapitre les éléments du modèle DIAL qui permettent de gérer le flux d'information en régulant l'enchaînement des messages afin de mener à bien l'interaction.

Les actes de langage qui sont proférés au cours des échanges ne sont pas isolés. Ils se répondent les uns aux autres. Les coups dialogiques sont des structures qui permettent de lier les actes de langage les uns aux autres.

Un système dialectique multi-agents est un dispositif formel au travers duquel deux agents argumentatifs jouent pour vérifier la validité d'une hypothèse. Ces deux joueurs échangent des coups en présence d'un troisième agent, appelé témoin. Le témoin est le responsable de la décision collective. Son arbitrage dépend non seulement des arguments avancés mais également de son estimation de la compétence relative de chacun des deux joueurs. Un dialogue est correct si le témoin s'est forgé

une conviction au terme des échanges. Il est probable que certains arguments soient passés sous silence. Un dialogue est complet si l'agent témoin s'est forgé au terme des échanges la même conviction que s'il disposait de tous les arguments du système.

Ce cadre formel permet ainsi d'évaluer les propriétés de deux protocoles particuliers : un protocole de demande d'information et un protocole de persuasion. Ce sont tous deux des protocoles à réponses uniques. Contrairement à une demande d'information, la charge de la preuve est partagée par les deux joueurs dans une persuasion. La terminaison des dialogues est garantie dans tous les cas de figure. À l'inverse, certaines hypothèses sur la situation initiale sont nécessaires pour garantir la correction et la complétude des dialogues.

Contrairement aux travaux de Henry Prakken [50, 49] (cf section 4.3.2), nous ne faisons aucune hypothèse concernant la conviction initiale des joueurs sur le thème. De cette manière, le système dialectique multi-agents proposé ici ne se limite pas aux discussions critiques qui sont des dialogues de persuasion particuliers. Il permet de jouer n'importe quel dialogue dont le but est la résolution d'un conflit.

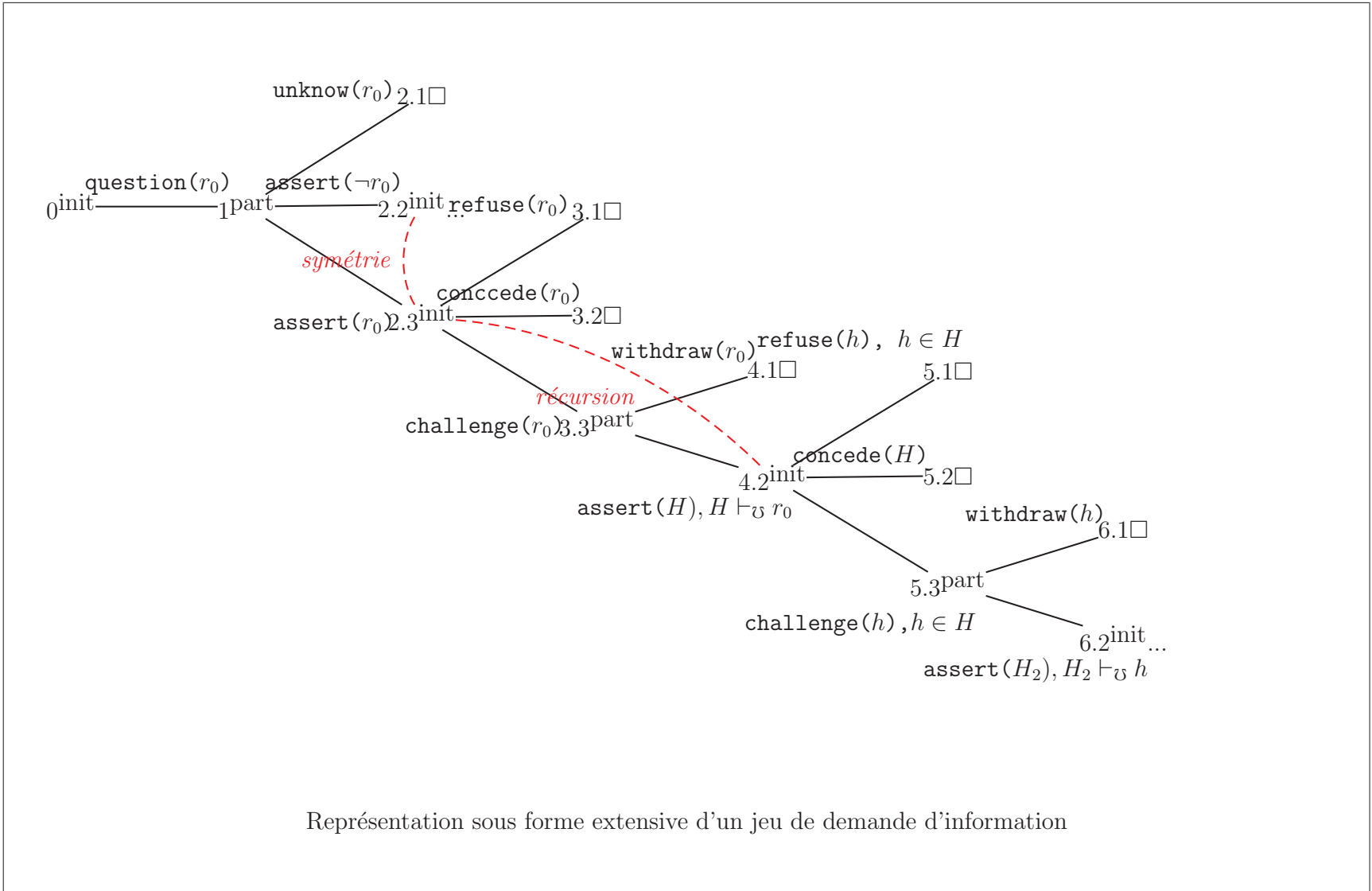


Fig. 9.6.1.1: Jeu de demande d'information

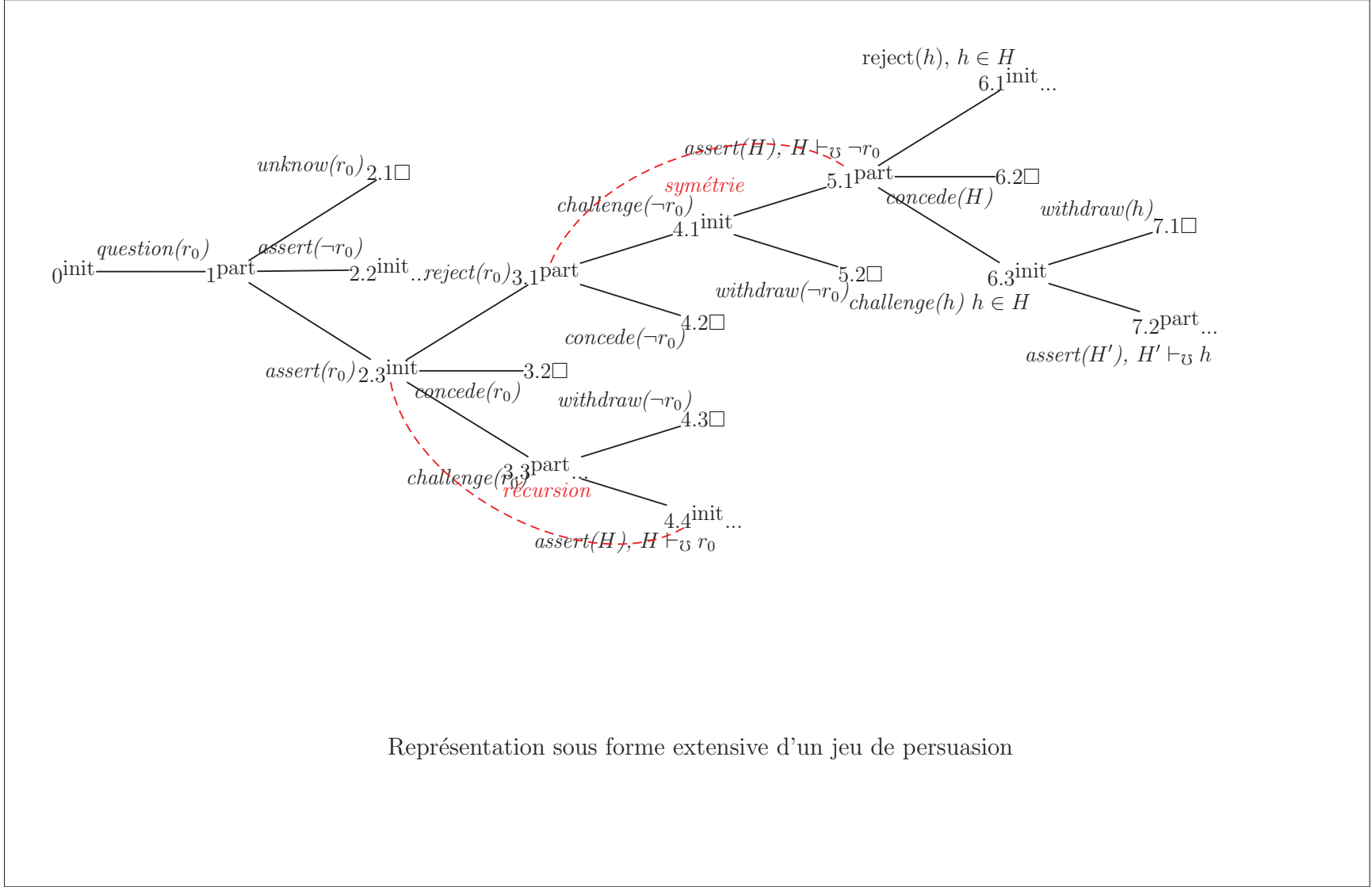


Fig. 9.6.2.1: Jeu de persuasion

Représentation sous forme extensive d'un jeu de persuasion

Chapitre 10

Conclusion du modèle

“In the original democracy, Athens, the agora was the marketplace, and more—it was where citizens met to talk, gossip, argue, size each other up, find the weak spots in political ideas by debating about them. But another kind of vision could apply to the use of the Net in the wrong ways, a shadow vision of a less utopian kind of place—the Panopticon.”

Howard Rheingold, 1993

Dans cette partie, nous avons proposé un modèle de dialogue au travers duquel les agents argumentatifs jouent et arbitrent pour aboutir à une décision collective. Celui-ci contribue à la formalisation d’une prise de décision concertée.

Nous avons proposé, dans le chapitre 7, une logique argumentative qui constitue le modèle de raisonnement des agents. Cette **logique argumentative à base de valeurs** permet de gérer les interactions entre des arguments contradictoires qui sont plus ou moins forts selon l’agent envisagé. Le langage logique sous-jacent permet d’exprimer des connaissances représentant l’état du monde. Plusieurs relations de priorité permettent de considérer simultanément les différentes échelles de valeurs auxquelles se réfèrent les agents.

Ayant défini le modèle de raisonnement des agents, nous avons présenté dans le chapitre 8 les éléments du modèle qui permettent aux agents d’une part d’échanger leurs connaissances et d’autre part de raisonner conjointement. Le **modèle d’agent argumentatif** proposé leur permet de justifier les hypothèses sur lesquelles ils s’engagent et de prendre en compte les engagements de leur interlocuteurs en fonction de la compétence estimée de l’agent qui a transmis cette information.

Ayant défini le modèle d’agent argumentatif, nous avons présenté dans le chapitre 9 les éléments du modèle qui permettent de gérer le flux d’information en régulant l’enchaînement des messages afin de mener à bien l’interaction. Dans un **système multi-agents dialectique**, un agent tiers, responsable de la décision finale, rend son arbitrage non seulement en fonction des arguments avancés mais également en fonction de son estimation de la compétence relative de chacun des deux joueurs. Le cadre ainsi circonscrit permet de garantir l’obtention d’un résultat au terme du dialogue et d’évaluer sa qualité.

Le modèle de dialogue entre agents proposé ici sert de base à l’outil informatique d’aide à la concertation présenté dans la partie suivante.

Troisième partie

Application

“Now is the time to shift our view of computers from communications medium to negotiation medium, from knowledge processing to interest processing.”

Carl Adam Petri

Chapitre 11

Système d'aide à la concertation

Sommaire

11.1 Introduction	186
11.2 Motivation	186
11.3 Principe	188
11.4 Logique argumentative	189
11.4.1 Représentation du problème	189
11.4.2 Expressions des préférences	190
11.4.3 Synthèse des jugements	192
11.4.4 Théorie argumentative décisionnelle	194
11.5 Système multi-assistants	197
11.6 Système dialectique multi-assistants	198
11.6.1 Jeu de soumission d'activité	198
11.6.2 Jeu de persuasion	198
11.7 Aide à la concertation	199
11.7.1 Élaboration conjointe d'un schéma argumentatif	199
11.7.2 Détection de conflit et de consensus	200
11.8 Implémentation	201
11.8.1 Architecture	201
11.8.2 Interface homme-machine	202
11.8.3 Modèle de raisonnement	203
11.8.4 Modèle d'agent	203
11.9 Synthèse	204

11.1 Introduction

Dans ce chapitre, nous présentons l'outil d'aide à la concertation élaboré en instanciant le modèle DIAL présenté dans la partie précédente. L'objectif d'un tel système consiste à fournir une véritable assistance aux acteurs dans un processus de prise de décision collective et débattue.

Un **système d'aide à la concertation** est un outil informatique interactif qui supporte une décision collective et débattue. Un tel système facilite la collaboration entre des acteurs qui travaillent à distance et de manière asynchrone. C'est un système de médiation [24, 25], *i.e.* outil informatique intelligent qui médiatise les débats dans le cadre de la démocratie dialogique [41] afin de faciliter l'identification et la résolution de conflits d'une manière rationnelle efficace et équitable.

Dans un premier temps, nous allons motiver notre approche (cf section 11.2). Le principe du système proposé ici est présenté dans la section 11.3. Le système s'appuie sur la méthode hiérarchique multicritère (cf section 11.4). Elle permet d'associer à chaque acteur un agent argumentatif (cf section 11.5). Chaque agent assiste un acteur et le représente dans des dialogues automatiques. Nous allons, dans la section 11.6, décrire le système dialectique multi-agents assistants. Il permet de fournir une véritable assistance à la concertation (cf section 11.7). Nous terminerons ce chapitre en présentant quelques éléments concernant l'implémentation qui a été proposée (cf section 11.8).

11.2 Motivation

La théorie des jeux fut introduite en 1940 par Oskar Morgenstern¹ et John von Neumann² [43]. S'appuyant sur un modèle économique du raisonnement, la théorie des jeux consiste à définir la rationalité d'un agent à l'aide d'une fonction de gain afin d'évaluer sa satisfaction vis à vis des différentes alternatives envisagées. Cette théorie fournit des critères pour mesurer la qualité des processus de prise de décision collective [56]. On peut notamment évaluer la qualité du résultat obtenu à l'issue de tels processus.

Afin de proposer un système d'aide à la concertation qui s'appuie sur un système multi-agents, une première approche peut consister à associer à chaque acteur un

¹Oskar Morgenstern (1902-1977) mathématicien et économiste américain d'origine allemande

²John von Neumann (1903-1957) mathématicien américain d'origine hongroise ayant apporté d'importantes contributions autant en physique quantique, qu'en analyse fonctionnelle, en théorie des ensembles, en informatique ou en sciences économiques.

agent qui le représente dans des marchandages (cf section 1.4.2.2). Dans cette hypothèse, l'acteur délègue une partie du processus de décision. L'agent signale à l'acteur qu'il assiste l'alternative préconisée à l'issue de chacune de ses conversations. Cette approche, permet de respecter le schéma cognitif et le système de valeur de chaque acteur. La vision et la compréhension d'un même problème sont soumises à la subjectivité de l'acteur. Toutefois, le théorème d'Arrow³ [34, 56] indique qu'il n'est pas envisageable de construire une fonction de choix social satisfaisante lorsqu'on dispose de plus de trois alternatives à un problème de décision.

Théorème n° 11.2.0.III [Théorème d'Arrow]

Soit $\mathcal{U}_A = \{ag_1, \dots, ag_i, \dots, ag_n\}$ un ensemble d'agents, chacun muni d'une relation de préférence \prec_i . Soit $A = \{a_1, a_2, \dots\}$ l'ensemble des alternatives considérées par les agents. Si $\|A\| \geq 3$ alors il n'existe pas de fonction de choix social (notée \prec_*) satisfaisant les propriétés suivantes :

1. **universalité**, *i.e.* la fonction de choix social doit être définie quelles que soient les préférences des agents ;
2. **non-dictature**, *i.e.* aucun agent ne doit pouvoir imposer ses préférences, indépendamment des préférences des autres :

$$\exists ag_i \in \mathcal{U}_A (a_1 \prec_i a_2 \Rightarrow a_1 \prec_* a_2);$$

3. **unanimité**, *i.e.* lorsque tous les agents ont les mêmes préférences, la fonction de choix social doit associer ces mêmes préférences à la société :

$$(\forall ag_i \in \mathcal{U}_A a_1 \prec_i a_2) \Rightarrow (a_1 \prec_* a_2)$$

4. **indifférence des alternatives non-pertinentes**, *i.e.* le classement relatif de deux alternatives ne doit dépendre que de leur position relative pour les agents et non du classement d'alternatives tierces.

Les économistes appellent préférences un préordre complet ou un ordre complet. Dans ce second cas de figure, on parle alors de préférences strictes (cf annexe C). La propriété selon laquelle la fonction de choix social doit être indifférente aux alternatives non-pertinentes implique en particulier que si l'on ne considère qu'un sous-ensemble d'alternatives, la fonction de choix social ne doit pas aboutir à un autre classement de ce sous-ensemble⁴.

³Kenneth Joseph Arrow (né le 23 août 1921) économiste américain qui a remporté avec John Hicks le prix Nobel de sciences économiques en 1972.

⁴Le résultat du second tour des élections présidentielles qui a eu lieu le 21 avril 2002 en est une illustration.

L'impossibilité de garantir l'existence d'un processus de décision collective de qualité a été démontrée dans [34]. Elle ne fait que confirmer la nécessité d'offrir une véritable assistance au débat.

11.3 Principe

Afin de fournir une véritable assistance au débat, le système doit agir en qualité de conseiller pour faciliter la recherche d'un accord mais laisser la décision finale aux acteurs. En supportant et en ne remplaçant pas le jugement humain, les utilisateurs sont au cœur de la problématique.

Le système représenté dans la figure 11.3.0.I s'inspire du système d'aide à la négociation proposé par Takayuki Ito et Toramatsu Shintani [30]. Il s'appuie sur un système multi-agents argumentatifs. Chaque agent argumentatif assiste un utilisateur et interagit avec les autres agents du système. Le système proposé ici fournit des fonctionnalités, d'une part pour l'élaboration collaborative des schémas argumentatifs et d'autre part, pour élucider les consistances et les inconsistances entre les préférences des acteurs et donc détecter les consensus et les conflits.

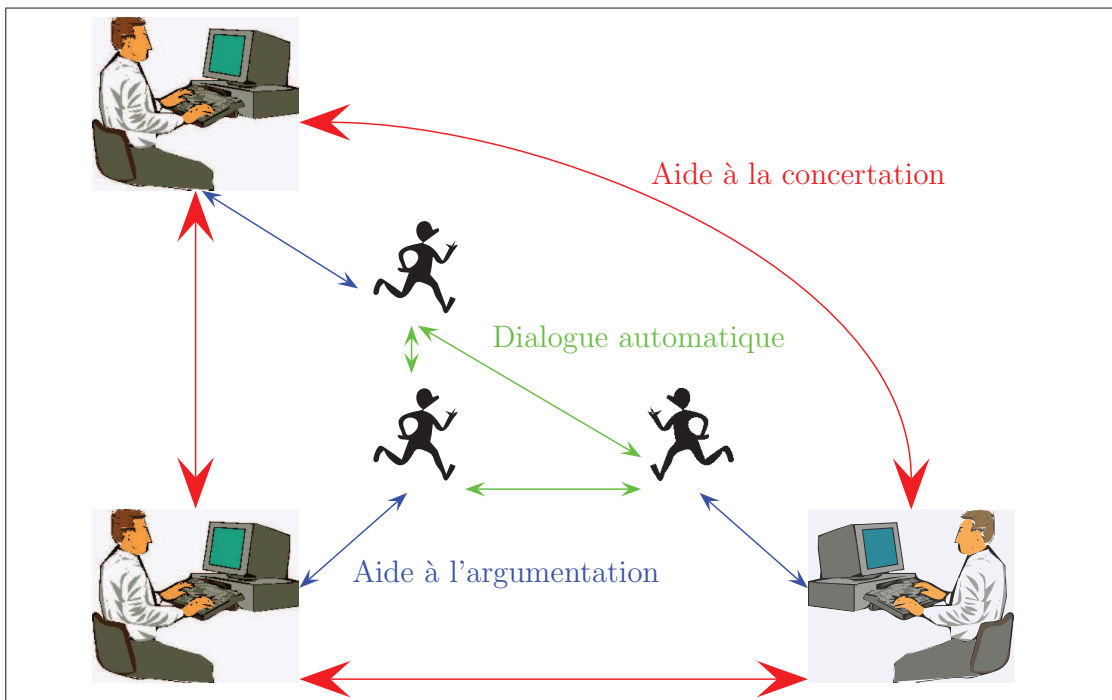


Fig. 11.3.0.I: Principe du système d'aide à la concertation

La logique argumentative à base de valeurs qui constitue le modèle de raisonnement des agents est présentée dans la section suivante.

11.4 Logique argumentative

Dans le domaine de la gestion territoriale, les outils d'aide à la décision utilisés par les acteurs sont majoritairement basés sur des méthodes d'Analyse Multi-Critères (*MCDA : Multi Criteria Decision Analysis*). Comme elle ne nécessite aucun prérequis quant aux connaissances scientifiques sous-jacentes, nous nous sommes intéressés à la **Méthode Hiérarchique Multicritère** (*Analytic Hierarchy Process*) proposée par Thomas L. Saaty⁵ [55, 54]. C'est un processus de décision puissant et flexible qui facilite l'expression des préférences et permet une prise de décision à l'aide de ces jugements qu'ils soient qualitatifs ou quantitatifs.

La Méthode Hiérarchique Multicritère (MHM) se décompose en trois étapes : la construction d'une représentation du problème (cf section 11.4.1), l'expression des préférences (cf section 11.4.2) et la synthèse des jugements (cf section 11.4.3). Cette méthode permet d'instancier le modèle de raisonnement des agents (cf section 11.4.4).

11.4.1 Représentation du problème

La MHM permet l'élaboration d'un schéma argumentatif [23, 53], pour obtenir une bonne représentation du problème. Pour parvenir à des connaissances précises, notre esprit structure la réalité complexe en diverses composantes, décompose ces dernières à leur tour et ainsi de suite, de façon hiérarchique. Dans une situation complexe, on construit une taxonomie des critères d'évaluation pour structurer la prise de décision.

La MHM structure le problème en une hiérarchie décisionnelle constituée d'éléments appelés activités. Le but du problème est identifié et décomposé en sous-problèmes appelés critères d'évaluation. Ces derniers permettent de comparer des alternatives. Les critères sont à leur tour décomposés en sous-critères et ainsi de suite jusqu'à obtenir les critères feuilles d'une taxonomie.

Définition n° 11.4.1.I [Hiérarchie décisionnelle]

- Une **hiérarchie décisionnelle** est un quadruplet $DH = \langle g, C, A, \triangleleft \rangle$ où :
- g est le but de la prise de décision ;
 - $C = \{c_0 = g, c_1, c_2, \dots\}$ est l'ensemble des critères d'évaluation ;
 - $A = \{a_1, a_2, \dots\}$ est l'ensemble des alternatives, *i.e.* des solutions possibles.
 $Act = C \cup A$ est appelé ensemble d'activités ;
 - \triangleleft est une relation de spécificité entre les activités, *c'est-à-dire* une relation d'ordre sur Act ($\triangleleft \subset Act \times Act$).

⁵Thomas L. Saaty professeur de mathématique à l'université de Pittsburgh.

La relation de spécificité est une relation d'ordre strict et partiel (cf annexe C). La relation $\text{act}_{k1} \triangleleft \text{act}_{k2}$ peut être interprétée de la manière suivante :

- soit act_{k1} est un sous-critère de act_{k2} ;
- soit act_{k1} est une alternative et act_{k2} un critère feuille.

La fermeture réflexo-transitive correspondante est notée \triangleleft^* .

L'exemple suivant permet d'illustrer cette définition.

Exemple n° 11.4.1.I [Représentation du problème] La figure 11.4.1.I illustre cette représentation du problème à travers l'exemple de la localisation d'ITER, la première installation expérimentale d'une centrale thermique à fusion nucléaire. Le but consiste à sélectionner le bon emplacement parmi trois alternatives, i.e. trois sites d'implantation qui sont évalués selon des critères organisés en taxinomie. Le but est décomposé en deux critères (c_1, c_2) eux-mêmes décomposés en sous-critères : c_3, c_4, c_5, c_6 et c_7 . La relation de spécificité est représentée en trait plein entre les critères et en trait pointillé entre les critères feuilles et les alternatives.

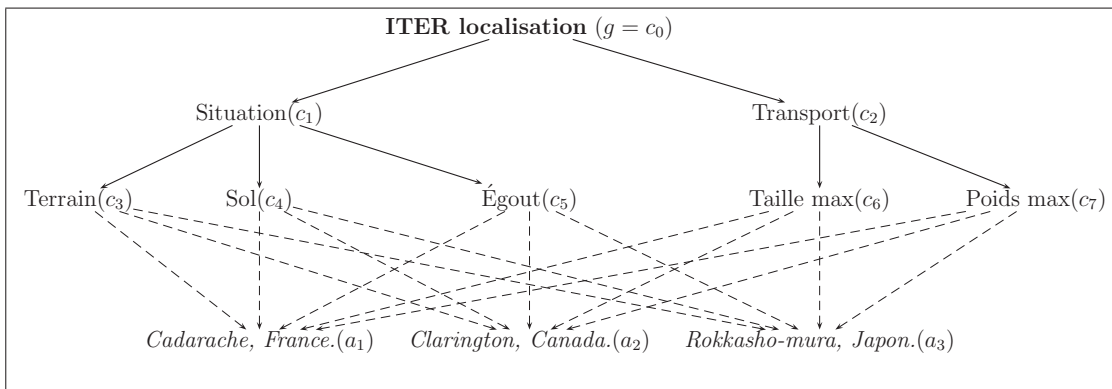


Fig. 11.4.1.I: Hiérarchie décisionnelle pour la localisation d'ITER

C'est à partir de cette représentation du problème que les préférences sont exprimées.

11.4.2 Expressions des préférences

La Méthode Hiérarchique Multicritère (MHM) s'appuie sur les sentiments et les jugements intuitifs qui sont probablement plus représentatifs de notre pensée, de nos comportements que la verbalisation que nous pouvons en faire.

La méthode consiste à comparer des couples d'activités similaires du même niveau, sur la base de l'activité parente et d'établir des distinctions entre les deux membres d'un couple en évaluant l'intensité des préférences de l'un par rapport à l'autre.

Comme l'illustre le tableau 11.4.2.I, on qualifie l'importance d'une activité par rapport à une autre : d'absolue (9), d'attestée (7), de déterminante (5), de faible (3)

ou d'égale (1). On peut estimer le poids relatif de l'activité i vis à vis de l'activité j étant donné un critère d'évaluation c_k dans le domaine de définition correspondant aux jugements qualitatifs de comparaison (1, 3, 5, 7, 9), à leurs valeurs intermédiaires (2, 4, 6, 8) mais également à leur réciproque ($\frac{1}{9}, \frac{1}{8}, \frac{1}{7}, \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}$).

Valeurs	Définitions
1	Importance égale
3	Faible importance
5	Importance déterminante
7	Importance attestée
9	Importance absolue
2,4,6,8	Valeurs intermédiaires
$\frac{1}{9}, \frac{1}{8}, \frac{1}{7}, \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}$	Valeurs réciproques

Tab. 11.4.2.I: Échelle de comparaisons binaires

L'ensemble des distinctions entre les activités similaires sur la base de l'activité parente peut être résumé dans une matrice de comparaisons binaires.

Définition n° 11.4.2.I [Matrice de comparaisons binaires]

Soit c_k le critère parent de n activités.

On appelle **matrice de comparaisons binaires** de c_k la matrice $A_{c_k} = (a_{ij})_{i,j \leq n}$ où chacun des éléments représente le poids relatif d'une activité act_i (noté w_i) vis à vis d'une activité act_j (noté w_j) : $\forall i, j \leq n \ a_{ij} = \frac{w_i}{w_j}$

Ainsi, le vecteur propre correspondant à la valeur propre maximale λ_{max} de A_{c_k} constitue le vecteur des priorités $W_{c_k} = (w_i)_{i \leq n}$. Le calcul de ce vecteur permet de connaître les priorités de chacune des activités ($w_i \in [0; 1]$) vis à vis du critère parent.

L'exemple suivant permet d'illustrer ces définitions.

Exemple n° 11.4.2.I [Établir ses préférences] *Considérons la hiérarchie décisionnelle de l'exemple 11.4.1.I. Le critère c_1 se décompose en trois sous-critères : c_3 , c_4 et c_5 . Le critère c_4 est quatre fois plus important que le critère c_3 et quatre fois plus important que le critère c_5 . Les propriétés supposées des éléments d'une matrice de comparaisons binaires comme l'identité, la réciprocity ou les valeurs par défaut permettent de réduire l'intervention de l'utilisateur. Ainsi, les critères c_3 et c_5 sont considérés d'importance égale. La matrice de comparaison binaire associée à ce critère est représentée dans le tableau 11.4.2.II. Le calcul du vecteur propre associé à la valeur propre maximale permet de déduire les priorités de chacun de ces critères.*

A_{c_1}	c_3	c_4	c_5	W_{c_1}
c_3	1	$\frac{1}{4}$	1	$w_1 = \frac{1}{6}$
c_4	4	1	4	$w_2 = \frac{2}{3}$
c_5	1	$\frac{1}{4}$	1	$w_3 = \frac{1}{6}$

Tab. 11.4.2.II: Matrice de comparaison binaire A_{c_1} et vecteur de priorités associé W_{c_1}

Les jugements ainsi exprimés doivent être synthétisés. On doit également vérifier leur cohérence.

11.4.3 Synthèse des jugements

Les préférences ne sont pas nécessairement cohérentes. Le ratio suivant permet d'évaluer cette cohérence :

Définition n° 11.4.3.I [Ratio de cohérence]

Soit $A_{c_k} = (a_{ij})_{i,j \leq n}$ une matrice de comparaisons binaires.

Le **ratio de cohérence** de la matrice A_{c_k} est défini par la formule

$RC_{A_{c_k}} = IC_{A_{c_k}} / CM_n$ où :

- $IC_{A_{c_k}} = \frac{\lambda_{max} - n}{(n-1)}$ est l'indice de cohérence de la matrice A_{c_k} ;
- CM_n est la cohérence moyenne, *i.e.* l'indice de cohérence moyen d'une matrice de comparaison binaire de taille n dont les valeurs sont aléatoires.

La cohérence moyenne dépend de la dimension de la matrice (n). Ces valeurs sont résumées dans le tableau 11.4.3.I. Ainsi, le ratio de cohérence est un taux ($RC_{A_{c_k}} \in [0; 1]$). Plus le ratio de cohérence est important, moins les préférences sont cohérentes. Le seuil d'acceptabilité est fixé à 5% ($RC_A \leq 5\%$) pour une matrice 3×3 , à 9% ($RC_A \leq 9\%$) pour une matrice 4×4 et à 10% ($RC_A \leq 10\%$) pour les matrices de plus grande taille.

n	1	2	3	4	5	6	7	8	9	10
CM_n	.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Tab. 11.4.3.I: Indices de cohérence pour des matrices aléatoires

Dans les limites de la cohérence ainsi définies, on peut déterminer les priorités relatives des différentes activités.

Définition n° 11.4.3.II [Hiérarchie décisionnelle valuée]

Une **hiérarchie décisionnelle valuée** est un triplet $VDH = \langle DH, I, J \rangle$ défini tq :

- $DH = \langle g, C, A, \triangleleft \rangle$ est une hiérarchie décisionnelle telle que nous l'avons définie précédemment ;
- $I : C \times C \rightarrow [0; 1]$ est une fonction qui détermine la priorité d'un critère vis à vis du critère parent : si $c_i \triangleleft c_j$ alors $I(c_i, c_j) = w_i \in W_{c_j}$;
- $J : A \times C \rightarrow [0; 1]$ est une fonction qui détermine la priorité d'une alternative vis à vis d'un critère feuille : $J(a_i, c_j) = w_i \in W_{c_j}$ avec c_j critère feuille.

L'extension de la fonction I (respectivement J) qui correspond à la fermeture transitive de la relation de spécificité (\triangleleft^*) est notée I^* (respectivement J^*). Ces fonctions sont définies de la manière suivante :

- I^* est une fonction qui détermine la priorité d'un critère vis à vis d'un critère ancêtre :

$$\begin{aligned} I^* : C \times C &\rightarrow [0; 1] \\ (c_i, c_j) &\mapsto I^*(c_i|c_j) = I(c_i|c_j) \text{ si } c_i \triangleleft c_j \\ &I^*(c_i|c_j) = I(c_i|c_k) \cdot I^*(c_k|c_j) \text{ si } c_i \triangleleft^* c_j \text{ avec } c_i \triangleleft c_k \end{aligned}$$

- J^* est une fonction qui détermine la priorité d'une alternative vis à vis d'un critère :

$$\begin{aligned} J^* : A \times C &\rightarrow [0; 1] \\ (a_i, c_j) &\mapsto J^*(a_i|c_j) = J(a_i|c_j) \text{ si } c_j \text{ est un critère feuille} \\ J^*(a_i|c_j) &= \sum_{c_k \triangleleft^* c_j, c_k \text{ critère feuille}} J(a_i|c_k) \cdot I^*(c_k|c_j) \end{aligned}$$

Ces fonctions rendent possible le calcul d'une alternative optimale a_m pour un critère c_l :

$$a_m \in \{a_n; \nexists a_o \in A \text{ tq } J^*(a_o, c_l) > J^*(a_n, c_l)\}$$

A fortiori une alternative a_m est optimale pour une hiérarchie décisionnelle valuée si elle est optimale vis à vis du but :

$$a_m \in \{a_n; \nexists a_o \in A \text{ s.a. } J^*(a_o, g) > J^*(a_n, g)\}$$

Cette modélisation de la décision synthétise les préférences sous la forme d'une fonction d'utilité :

$$\begin{aligned} u : A &\rightarrow [0; 1] \\ a_i &\mapsto u(a_i) = J^*(a_i|c_0) \end{aligned}$$

La fonction d'utilité associée à chaque alternative un gain.

Exemple n° 11.4.3.I [Synthèse des jugements] *Considérons la hiérarchie décisionnelle de l'exemple 11.4.1.1. Comme préconisé dans la section 11.4.2, les préférences ont été exprimées à l'aide de comparaisons binaires entre des activités similaires de même niveau. Les priorités des différentes activités vis à vis de l'activité parente sont représentées dans la figure 11.4.3.1. Le gain associé au site d'implantation français est calculé de la manière suivante :*

$$\begin{aligned}
u(a_1) &= J(a_1|c_0) \\
&= J(a_1|c_3) \cdot I^*(c_3|c_0) + J(a_1|c_4) \cdot I^*(c_4|c_0) + J(a_1|c_5) \cdot I^*(c_5|c_0) \\
&\quad + J(a_1|c_6) \cdot I^*(c_6|c_0) + J(a_1|c_7) \cdot I^*(c_7|c_0) \\
&= \frac{2}{3} \cdot I^*(c_3|c_0) + \frac{1}{6} \cdot I^*(c_4|c_0) + \frac{1}{6} \cdot I^*(c_5|c_0) + \frac{2}{3} \cdot I^*(c_6|c_0) + \frac{1}{6} \cdot I^*(c_7|c_0) \\
&= I(c_1|c_0) \cdot \left[\frac{2}{3} \cdot I(c_3|c_1) + \frac{1}{6} \cdot I(c_4|c_1) + \frac{1}{6} \cdot I(c_5|c_1) \right] \\
&\quad + I(c_2|c_0) \cdot \left[\frac{2}{3} \cdot I(c_6|c_2) + \frac{1}{6} \cdot I(c_7|c_2) \right] \\
&= \frac{1}{2} \cdot \left[\frac{2}{3} \cdot \frac{1}{6} + \frac{1}{6} \cdot \frac{2}{3} + \frac{1}{6} \cdot \frac{1}{6} \right] + \frac{1}{2} \cdot \left[\frac{2}{3} \cdot \frac{3}{4} + \frac{1}{6} \cdot \frac{1}{4} \right] \\
&\simeq 0,3958
\end{aligned}$$

Les gains associés aux sites d'implantation canadien et japonais sont calculés de manière similaire :

$$\begin{aligned}
u(a_2) &= J(a_2|c_0) \simeq 0,3333 \\
u(a_3) &= J(a_3|c_0) \simeq 0,2708
\end{aligned}$$

On constate que le site français est optimal pour cette hiérarchie décisionnelle évaluée.

En résumé, cette méthodologie d'aide à la décision permet à un acteur de représenter un problème, d'exprimer ses préférences à partir de cette représentation et de synthétiser ses appréciations en mesurant leur cohérence. C'est à partir de cette méthodologie que le modèle de raisonnement des agents est construit.

11.4.4 Théorie argumentative décisionnelle

L'aspect quantitatif de la MHM permet de doter les agents d'un modèle de raisonnement économique. La rationalité d'un agent est alors définie à l'aide d'une fonction de gain. Nous avons montré dans la section 11.2 que cette approche n'est pas adaptée. L'approche qualitative de la MHM permet de doter les agents d'un modèle de raisonnement argumentatif. La rationalité d'un agent est alors définie à l'aide d'une

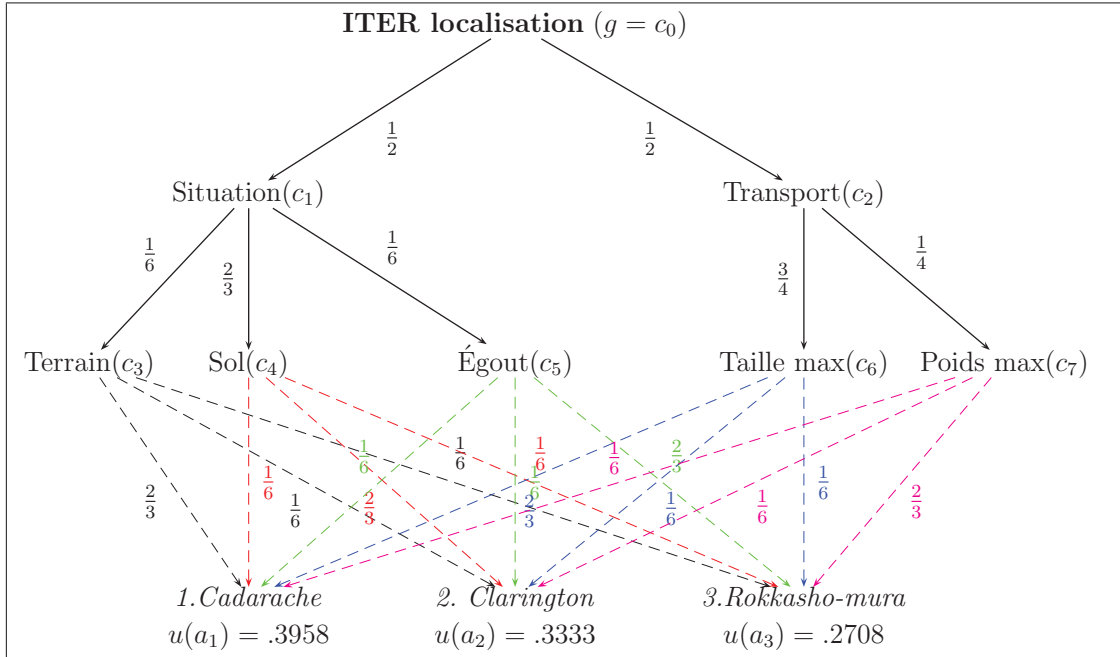


Fig. 11.4.3.I: Hiérarchie décisionnelle valuée pour la localisation d'ITER

théorie argumentative décisionnelle. Nous montrons dans cette section qu'une hiérarchie décisionnelle valuée est une théorie argumentative mono-agent particulière (cf chapitre 7).

Une hiérarchie décisionnelle est une représentation du problème. Cette connaissance peut être formalisée à l'aide d'un langage de logique classique. Étant donné une hiérarchie décisionnelle, le **langage décisionnel** associé (noté \mathcal{L}^{DH}) contient un ensemble d'atomes (noté Act) qui correspondent aux activités. On doit pouvoir raisonner à partir de ces connaissances. À cet effet, on utilise un système de preuves (cf Annexe A). \vdash^{DH} dénote la déduction classique.

Ces connaissances doivent être rassemblées. Nous utilisons à cet effet une **théorie décisionnelle** (notée \mathcal{T}^{DH}), *i.e.* l'ensemble de règles suivantes :

- $r_g : g \leftarrow$ est la règle qui correspond au but de la hiérarchie décisionnelle ;
- $r_e(\text{act}_l, \text{act}_{l'}) : \text{act}_l \leftarrow \neg \text{act}_{l'}$ sont les règles qui correspondent aux exclusions mutuelles entre activités similaires ($\text{act}_l, \text{act}_{l'} \triangleleft \text{act}_k$) ;
- $r_s(\text{act}_l, \text{act}_k) : \text{act}_l \leftarrow \text{act}_k$ sont les règles de spécialisation qui correspondent à la fermeture transitive de la relation de spécificité ($\text{act}_l \triangleleft^* \text{act}_k$).

Les préférences peuvent être exprimées sous la forme d'une relation de priorité (cf définition 7.2.0.IV). Cette **relation de priorité décisionnelle** (notée \ll_{VDH}) est définie tq :

- $\forall r \in \mathcal{T}^{DH} r \ll_{VDH} r_g$, *i.e.* le but est prioritaire ;
- $\forall r \in \mathcal{T}^{DH} \forall \text{act}_l, \text{act}_{l'} \in \text{Act} r \ll_{VDH} r_e(\text{act}_l, \text{act}_{l'})$, *i.e.* une heuristique selon

laquelle les exclusions mutuelles entre activités sont prioritaires ;

- $r_s(\text{act}_{l_1}, \text{act}_{k_1}) \ll_{\text{VDH}} r_s(\text{act}_{l_2}, \text{act}_{k_2})$ est équivalent à $I/J^*(\text{act}_{l_1}, \text{act}_{k_1}) \leq I/J^*(\text{act}_{l_2}, \text{act}_{k_2})$, *i.e.* une heuristique selon laquelle la priorité d'une règle de spécialisation est proportionnelle à la priorité de l'alternative vis à vis de l'alternative ancêtre.

On remarque que la relation de priorité décisionnelle est une relation d'ordre total et large.

On peut associer à une hiérarchie décisionnelle valuée une théorie argumentative décisionnelle.

Définition n° 11.4.4.I [Théorie argumentative décisionnelle]

Soit $\text{VDH} = \langle \text{DH}, I, J \rangle$ une hiérarchie décisionnelle valuée. La **théorie argumentative décisionnelle** associée est un quadruplet

$\mathcal{AT}^{\text{VDH}} = \langle \mathcal{T}^{\text{DH}}, V^{\text{VDH}}, \text{promote}, \ll_{\text{VDH}} \rangle$ où :

- \mathcal{T}^{DH} est une théorie décisionnelle telle que nous l'avons définie précédemment ;
- V^{VDH} est un ensemble fini de valeurs $\{v_1, \dots, v_t\}$;
- $\text{promote}_{\text{VDH}} : \mathcal{T}^{\text{DH}} \rightarrow V^{\text{VDH}}$ est une fonction qui met en relation les règles et les valeurs. Pour tous $r \in \mathcal{T}^{\text{DH}}$, $\text{promote}_{\text{VDH}}(r) = v$;
- \ll_{VDH} est la relation de priorité décisionnelle associée telle que nous l'avons définie précédemment.

Cette théorie argumentative décisionnelle n'est rien d'autre qu'une théorie argumentative mono-agent (cf définition 7.2.0.V). L'ensemble des arguments subjectivement acceptables selon cette théorie sera noté \mathcal{S}_{VDH} .

On peut ainsi facilement démontrer qu'une alternative optimale pour un critère est supportée par un argument acceptable.

Théorème n° 11.4.4.I [Alternative optimale/Argument acceptable]

Soit $\text{VDH} = \langle \text{DH}, I, J \rangle$ une hiérarchie décisionnelle valuée et

$\mathcal{AT}^{\text{VDH}} = \langle \mathcal{T}^{\text{DH}}, V^{\text{VDH}}, \text{promote}_{\text{VDH}}, \ll_{\text{VDH}} \rangle$ la théorie argumentative correspondante.

$$a_m \in \{a_n; \nexists a_o \in A \text{ tq } J^*(a_o, c_k) > J^*(a_n, c_k)\}$$

$$\Leftrightarrow \exists A \in \mathcal{S}_{\text{VDH}} \text{ conclusion}(A) = a_m \leftarrow c_k$$

Démonstration n° 11.4.4.I [Alternative optimale/Argument acceptable]

Trivial. □

Ayant défini le modèle de raisonnement des agents, nous allons présenter dans la section suivante les éléments du système d'aide à la concertation qui permettent aux agents d'échanger des informations.

11.5 Système multi-assistants

Le système multi-agents argumentatifs déployé ici est similaire à celui proposé dans le chapitre 8. Il permet aux agents d'une part d'échanger leurs connaissances et d'autre part de raisonner conjointement. De plus, les agents assistants peuvent négocier la théorie commune.

Les agents assistants, qui disposent de leurs propres hiérarchies décisionnelles, enregistrent non seulement les engagements pris par leurs interlocuteurs mais également les activités qu'ils prennent en considération conjointement avec leurs interlocuteurs. Chacun des agents assistants est conforme à la définition suivante :

Définition n° 11.5.0.II [Agent assistant]

Un agent assistant $ag_i \in \mathcal{U}_A$ est défini par un septuplet

$ag_i = \langle \mathcal{T}^{DH_i}, V^{VDH_i}, \ll_{VDH_i}, promote_i, \cup_{j \neq i} CS_j^i(t), \cup_{j \neq i} CSAct_j^i(t), \prec_i \rangle$ où :

- $\mathcal{T}^{DH_i} \subseteq \mathcal{T}^{DH_{\mathcal{U}_A}}$ est la théorie décisionnelle personnelle de l'agent, i.e. la théorie associée à la hiérarchie décisionnelle de l'agent ;
- V^{VDH_i} est un ensemble fini de valeurs personnelles qui constituent les valeurs considérées par l'agent ;
- $promote_i : \mathcal{T}^{DH_i} \rightarrow V^{VDH_i}$ est une fonction qui met en relation les règles personnelles et les valeurs personnelles ;
- \ll_{VDH_i} est la relation de priorité décisionnelle associée à l'agent ;
- $CS_j^i(t)$ est le tableau d'engagement constitué des engagements pris avant ou à l'instant t dont l'agent ag_i est le créancier et l'agent ag_j le débiteur ;
- $CSAct_j^i(t) \subset \mathcal{T}^{DH_i}$ est le tableau des activités communes avec celles de l'agent ag_j ;
- \prec_i est la relation de réputation associée à l'agent.

Puisque chaque agent ag_i représente un utilisateur, les schémas argumentatifs (DH_i) et leur valuation (VDH_i) diffèrent à l'exception du but qui est partagé par l'ensemble des utilisateurs. C'est la raison pour laquelle chaque agent a sa propre théorie décisionnelle personnelle \mathcal{T}^{DH_i} et sa propre relation de priorité décisionnelle.

Au cours des dialogues, les agents échangent et partagent des activités. Le tableau d'activités, noté $CSAct_j^i$, est le sous-ensemble de Act_i que l'agent ag_j partage avec l'agent ag_i . Il est initialement restreint au but et conserve la trace des activités partagées au cours des dialogues, que celles-ci soient des critères CSC_j^i ou des alternatives CSA_j^i .

Ayant défini le modèle d'agent assistant, nous allons présenter dans la section suivante les éléments du système d'aide à la concertation qui permettent de gérer le

flux d'information en régulant l'enchaînement des messages afin de mener à bien les interactions.

11.6 Système dialectique multi-assistants

Le système dialectique multi-agents argumentatifs déployé ici est similaire à celui proposé dans le chapitre 9. Le cadre ainsi circonscrit permet de garantir l'obtention d'un résultat au terme des dialogues.

Nous décrivons ici deux jeux de dialogue. Le premier permet de modifier l'ensemble des activités communes. Le second permet l'élucidation des conflits et des consensus.

11.6.1 Jeu de soumission d'activité

Le but de tels dialogues consiste à révéler une activité prise en compte par l'initiateur et à la suggérer au partenaire.

La figure 11.6.1 représente deux jeux de soumission d'activité sous forme normale extensive. Le jeu de soumission de critère représenté à gauche permet à l'initiateur de suggérer au partenaire un nouveau critère. Le jeu de soumission d'alternative représenté à droite permet à l'initiateur de suggérer au partenaire une nouvelle alternative.

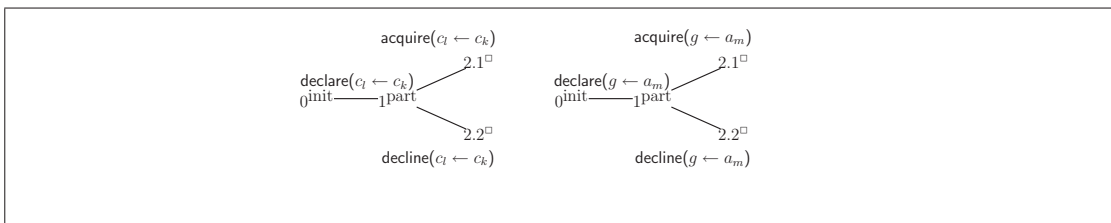


Fig. 11.6.1.I: Jeux de soumission d'activité sous forme normale extensive

Un jeu de dialogue défini de manière similaire permet à l'initiateur de signaler au partenaire qu'il se désolidarise d'une activité.

L'ensemble des activités ainsi partagées permet aux agents assistants de justifier leur choix lorsqu'ils sont confrontés les uns aux autres dans des persuasions.

11.6.2 Jeu de persuasion

Le but de tels dialogues consiste à révéler les positions des agents assistants vis à vis du but commun.

La figure 9.6.2.I représente un jeu de persuasion sous forme normale extensive. Le protocole utilisé est celui présenté dans la section 9.6.2. De plus une hypothèse est transmise par l'agent ag_i à l'agent ag_j si les atomes qui la constituent sont dans le tableau des activités communes $C\text{SAct}_j^i$.

Les jeux de dialogue présentés ici permettent de fournir une véritable assistance à la concertation.

11.7 Aide à la concertation

Le système d'aide à la concertation permet d'une part aux acteurs d'élaborer collaborativement leur schéma argumentatif et d'autre part, d'examiner les consistances et les inconsistances entre leurs préférences.

11.7.1 Élaboration conjointe d'un schéma argumentatif

Comme le système HERMES [33] et Zeno [23], le système d'aide à la concertation proposé ici permet d'élaborer conjointement un schéma argumentatif. Étant donné que les expertises sont distribuées, le processus de partage des activités proposé ici, permet de créer une hiérarchie décisionnelle commune en respectant de plus le schéma cognitif des acteurs.

Quand un agent ag_i dispose d'un nouveau sous-critère c_l de c_k dans sa hiérarchie décisionnelle DH_i , il met à jour la hiérarchie décisionnelle évaluée VDH_i et diffuse cette activité dans le système (cf algorithme 11.7.1). L'agent ag_i débute un dialogue de soumission d'activité avec l'ensemble des interlocuteurs qui possèdent le critère c_k dans leur hiérarchie décisionnelle.

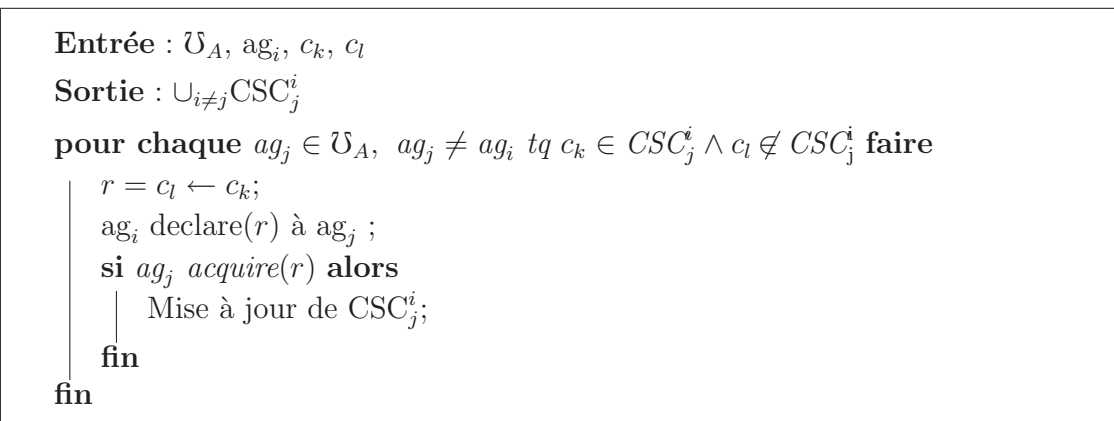


Fig. 11.7.1.I: Diffusion d'un nouveau critère c_l

L'agent ag_j qui reçoit la déclaration d'une hypothèse $r : c_l \leftarrow c_k$ peut répondre par une acquisition si le critère c_l est dans sa hiérarchie décisionnelle DH_j . Dans le cas contraire, il suggère ce nouveau critère à son utilisateur. Si ce dernier décide de ne pas prendre ce critère en compte, l'agent ag_j décline la proposition. Sinon l'agent ag_j s'acquiesce de cette proposition. Dans ce dernier cas, l'agent ag_j doit à son tour diffuser cette modification dans le système.

Le processus de partage des alternatives est très similaire. Une nouvelle alternative est suggérée à l'ensemble des acteurs par l'intermédiaire du système. Les utilisateurs qui intègrent une nouvelle activité dans leur schéma argumentatif doivent la valuer. De manière similaire, les acteurs peuvent se désolidariser d'une activité. Ce système d'aide à la concertation permet de négocier une représentation commune du problème. Tous les agents partagent le même but mais chacun d'eux a son propre ensemble d'activités : alternatives et critères. L'ensemble des activités peut se dilater ou se réduire au cours du débat.

Exemple n° 11.7.1.I [Élaboration conjointe d'un schéma argumentatif]

Considérons trois acteurs qui souhaitent débattre de la localisation d'ITER. La hiérarchie décisionnelle conjointe est représentée dans la figure 11.7.1.II. La hiérarchie décisionnelle commune est initialement constituée du but et des deux sites d'implantation japonais et canadien. Le premier acteur, représenté en rouge, prend en considération le critère commun c_0 et les critères c_2 , c_6 et c_7 . Il évalue les deux alternatives communes : a_2 et a_3 . Le second acteur, représenté en vert, considère le but commun g ainsi que deux sous-critères c_1 et c_2 . Il prend également en considération deux sous-critères de c_2 : c_6 et c_7 . Le troisième acteur, représenté en bleu, prend en considération les critères c_0 , c_1 , c_3 , c_4 et c_5 . Il évalue trois alternatives.

C'est parmi les activités partagées par les acteurs que les conflits et les consensus sont détectés.

11.7.2 Détection de conflit et de consensus

Comme les jugements sont subjectifs, le système fournit des fonctionnalités qui permettent d'examiner les consistances et les inconsistances entre les préférences des utilisateurs.

Considérons deux agents assistants. Un critère consensuel est un critère qu'ils partagent et pour lequel une des alternatives optimales est commune. À l'inverse, un critère conflictuel est un critère qu'ils partagent et pour lequel les alternatives optimales sont toutes différentes. Une persuasion entre ces deux agents permet d'identifier le conflit ou le consensus majeur qu'ils partagent. On peut facilement démontrer qu'une persuasion prend fin : soit lorsqu'un des critères consensuels parmi les plus

généraux est atteint, soit lorsque un critère conflictuel parmi les plus spécifiques est atteint. À l'issue d'une telle persuasion :

- soit un des joueurs concède une hypothèse ($\text{concede}(a_m \leftarrow c_k)$). Les conditions d'énonciation des deux derniers actes permettent de déduire que, selon ces deux joueurs, a_m est une alternative optimale pour c_k . En d'autres termes, ce critère est consensuel. Aucun critère plus général n'est consensuel. Dans le cas contraire, une précédente concession aurait mis un terme au dialogue.
- soit un des joueurs abandonne une hypothèse ($\text{withdraw}(c_k \rightarrow a_m)$). Les conditions d'énonciation des affirmations omises et la tactique d'énonciation de la mise en doute a mené l'exploration de la hiérarchie décisionnelle commune vers un critère conflictuel. Si ce critère avait pu être justifié par un sous-critère, cela aurait été le cas. En conséquence, ce critère conflictuel est l'un des plus spécifiques.

L'exemple suivant permet d'illustrer ce processus.

Exemple n° 11.7.2.I [Détection de conflit et de consensus] *Considérons une persuasion entre deux agents assistants. Les hiérarchies décisionnelles évaluées de l'initiateur et du partenaire sont représentées respectivement à gauche et à droite de la figure 11.7.2.I. Outre la hiérarchie décisionnelle commune initialement constituée du but et des deux alternatives a_1 et a_2 , ces deux agents partagent un ensemble d'activités communes : c_1 , c_3 et c_4 . Ce sont ces critères qui sont utilisés lors de la persuasion. La persuasion entre ces deux agents est représentée dans le tableau 11.7.2.II. Le critère atteint à l'issue du dialogue est c_4 . On vérifie que ce critère est un des critères conflictuels les plus spécifiques. Si les rôles conventionnels des joueurs avaient été intervertis, le critère atteint à l'issue du dialogue aurait été c_3 . Ce critère est également un des critères conflictuels les plus spécifiques. c_4 et c_3 sont tous deux à l'origine du différent entre les deux acteurs.*

Le jeu de persuasion consiste en une exploration de la hiérarchie décisionnelle commune, une recherche en profondeur pour l'analyse et la confrontation des préférences.

11.8 Implémentation

Ayant présenté le système d'aide à la concertation, nous allons fournir ici quelques éléments quant à l'implémentation qui en a été proposée. Nous allons tout d'abord présenter l'architecture du système (cf section 11.8.1) puis décrire brièvement chacun de ces composants (cf sections 11.8.2, 11.8.3, 11.8.4).

11.8.1 Architecture

L'architecture du système d'aide à la concertation présentée ici s'appuie sur une architecture multi-agents.

L'architecture du système est représentée dans la figure 11.8.1.I. Nous avons représenté dans cette figure deux agents. Ils communiquent à l'aide d'une infrastructure de communication (*middleware*). À cette intention, nous utilisons l'infrastructure de communication DEMAS (*Distributed Environment for Multi-Agent System*) qui permet de gérer l'exécution répartie, l'identification des agents et la communication par envoi de message. DEMAS est l'un des modules de la plateforme MAST⁶ (*Multi-Agent System Toolkit*) développée au sein de notre équipe de recherche.

Chaque utilisateur interagit avec son agent assistant au travers d'une Interface Homme-Machine (IHM). À cette intention, nous avons développé une application intitulée JAHP (*Java Analytic Hierarchy Process*). Ce logiciel qui fournit des fonctionnalités pour l'aide à la concertation est décrit dans la section 11.8.2. Il permet notamment de construire la hiérarchie décisionnelle évaluée. Cette dernière lorsqu'elle est traduite en théorie argumentative décisionnelle permet d'instancier le modèle de raisonnement de l'agent. Afin de l'implémenter, nous avons développé une application intitulée PVAL (*Prolog Value-based Argumentative Logic*). Ce logiciel est décrit dans la section 11.8.3. C'est à partir de ce modèle de raisonnement que le modèle d'agent argumentatif est implémenté. À cette intention, PADIAL (*Prolog Agent for DIAL*), qui décrit dans la section 11.8.4, interprète les coups reçus et énonce éventuellement une réponse en fonction des arguments disponibles et de la convention utilisée. De tels agents dialoguent.

11.8.2 Interface homme-machine

JAHP (*Java Analytic Hierarchy Process*) est une application que nous avons développée en Java. Ce logiciel implémente les éléments d'Interface Homme-Machine (IHM) du système d'aide à la concertation.

JAHP est munie d'une interface graphique développée en JFC/Swing qui permet de mettre en œuvre la méthode hiérarchique multicritère. La figure 11.8.2.I représente une capture d'écran de cette interface. La hiérarchie des critères est représentée dans le panneau supérieur gauche. Les alternatives sont représentées dans le panneau inférieur droit. Le bouton de contrôle au centre permet d'évaluer l'importance relative d'une activité par rapport à une autre. Les priorités de ces activités ainsi que le ratio de cohérence correspondant sont affichés dans le panneau supérieur droit. Les priorités des alternatives vis à vis du but sont calculées et affichées dans le panneau inférieur gauche.

C'est à partir de cette interface que le modèle de raisonnement de l'agent est instancié.

⁶Logiciel libre sous licence Apache 2.0 : <http://www.emse.fr/~vercouter/mast/index.html>

11.8.3 Modèle de raisonnement

PVAL (*Prolog Value-based Argumentative Logic*) est une application que nous avons implémentée en SWI-Prolog⁷. Ce logiciel implémente le modèle de raisonnement des agents.

PVAL est constitué d'un ensemble de prédicats qui permettent de calculer les arguments en faveur ou en défaveur d'une conclusion et de déterminer leur acceptabilité. Ce logiciel s'inspire du cadre général d'argumentation GORGIAS⁸ proposé par Neophytos Demetriou. La hiérarchie décisionnelle évaluée construite à l'aide de JAHP est traduite en une théorie argumentative mono-agent conformément à la syntaxe de PVAL. La figure 11.8.3.I représente la théorie argumentative de l'agent ag_1 (cf exemple 7.2.0.I) telle qu'elle est implémentée en PVAL. Les règles de la théorie personnelle sont énumérées dans la partie supérieure du programme. Chacune de ces règles est en relation avec une priorité. Les relations de priorités entre ces valeurs sont énumérées dans la partie inférieure du programme. PVAL permet de déterminer si un littéral est la conclusion d'un argument. Ce prédicat est noté `supported(+Literal)`. PVAL permet également de déterminer si un littéral est la conclusion d'un argument acceptable. Ce prédicat est noté `convinced(+Literal)`.

C'est à partir de ce modèle de raisonnement que le modèle d'agent argumentatif est implémenté.

11.8.4 Modèle d'agent

PADIAL (*Prolog Agent for DIAL*) est une application que nous avons implémentée en SWI-Prolog. Ce logiciel implémente le modèle d'agent argumentatif.

PADIAL est muni d'un ensemble de prédicats qui permettent de mettre à jour les tableaux d'engagements et de calculer la réponse à un coup en fonction des arguments disponibles, des conditions d'illocution et des stratégies d'énonciation mais également, en fonction des règles de séquence et des stratégies conventionnelles. La figure 11.8.4.I représente les règles qui permettent à un agent de répondre à une question. La règle de « Question/Réponse » ainsi que les tactiques conventionnelles associées sont représentées dans la partie supérieure du programme. Les règles relatives à la tactique d'énonciation d'une affirmation sont énumérées dans la partie inférieure du programme.

⁷La plus connue des implémentations standards et libres de Prolog : <http://www.swi-prolog.org/>

⁸Logiciel libre disponible à l'adresse suivante : <http://www2.cs.ucy.ac.cy/~nkd/gorgias/>

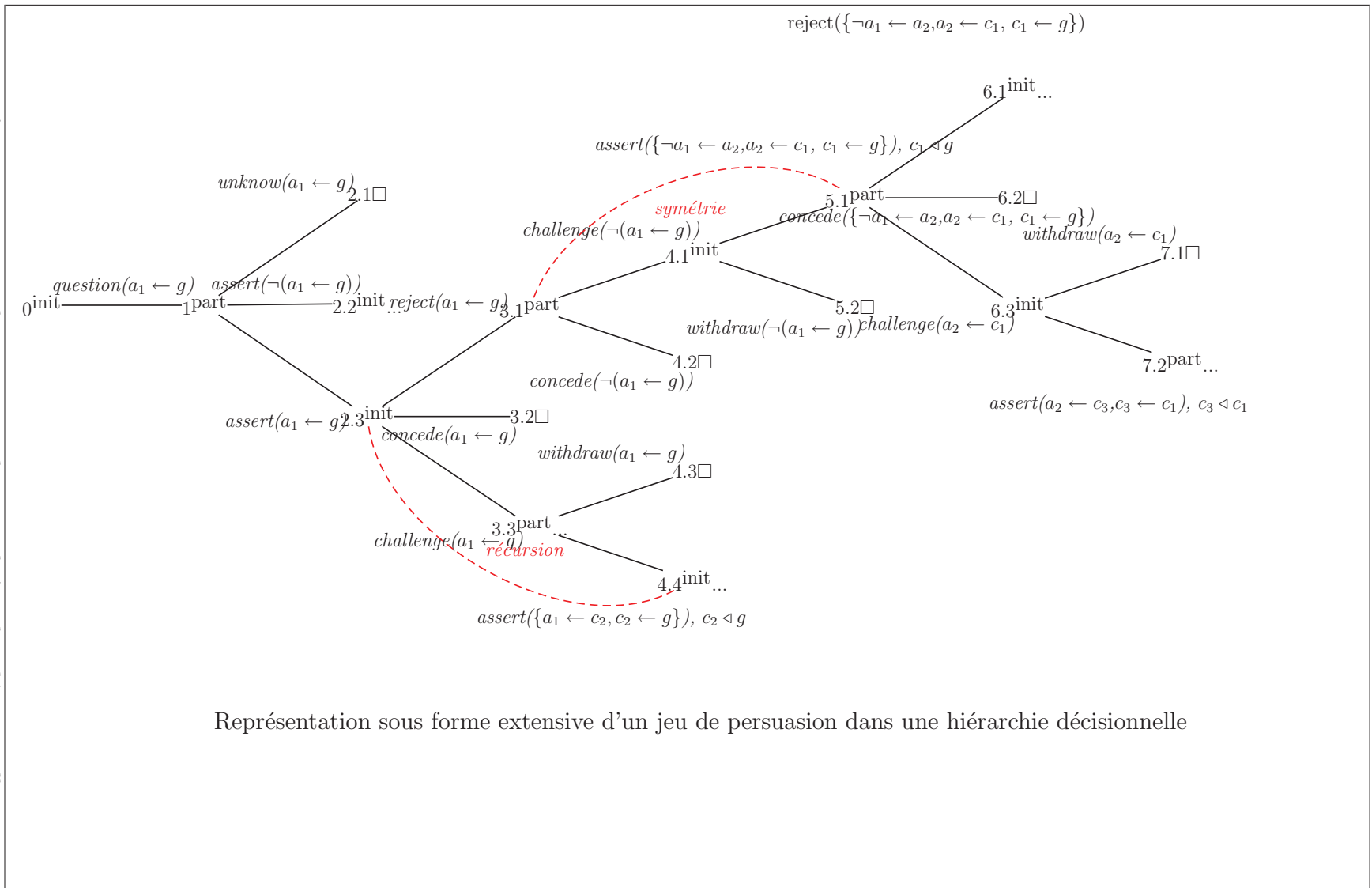
11.9 Synthèse

Dans ce chapitre, nous avons présenté l'outil d'aide à la concertation élaboré à l'aide du modèle DIAL.

Un système d'aide à la concertation est un outil informatique interactif qui supporte une décision collective et débattue. Un tel système facilite la collaboration entre des acteurs qui travaillent à distance et de manière asynchrone. Afin de fournir une véritable assistance au débat, le système agit en qualité de conseiller pour faciliter la recherche d'un accord mais laisse la décision finale aux acteurs. En supportant et en ne remplaçant pas le jugement humain, les utilisateurs sont au cœur de la problématique.

Le système d'aide à la concertation proposé ici s'appuie sur un système multi-agents argumentatifs. Chaque agent argumentatif assiste un utilisateur et interagit avec les autres agents du système. D'une part, le système permet de représenter un problème, d'exprimer des préférences à partir de cette représentation et de synthétiser ces appréciations en mesurant leur cohérence. D'autre part, le système fournit des fonctionnalités pour l'élaboration collaborative des schémas argumentatifs et pour élucider les consistances et les inconsistances entre les préférences des acteurs et donc détecter les consensus et les conflits.

Fig. 11.6.2.1: Jeu de persuasion dans une hiérarchie décisionnelle



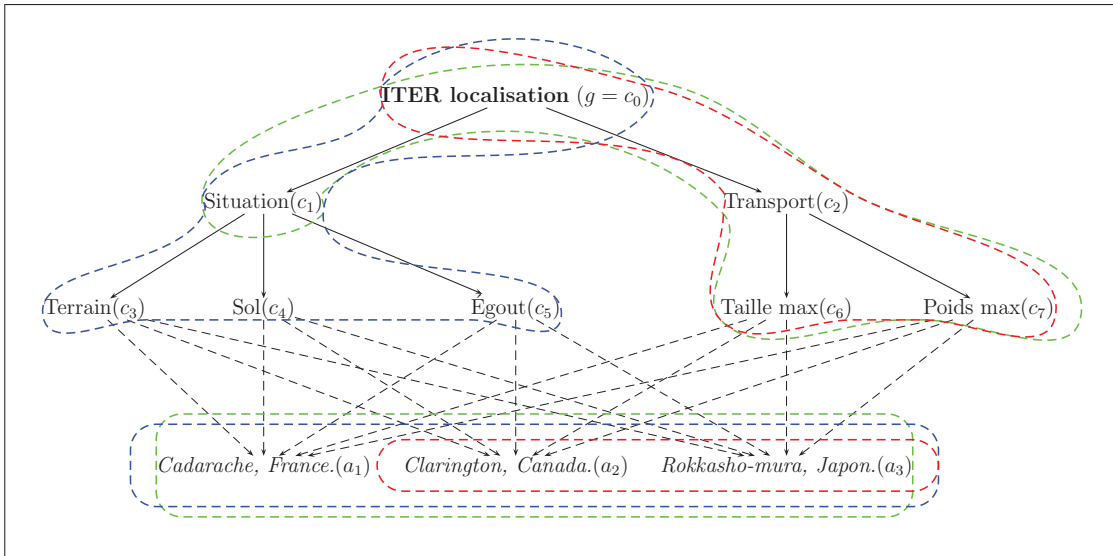


Fig. 11.7.1.II: Hiérarchie décisionnelle conjointe pour la localisation d'ITER

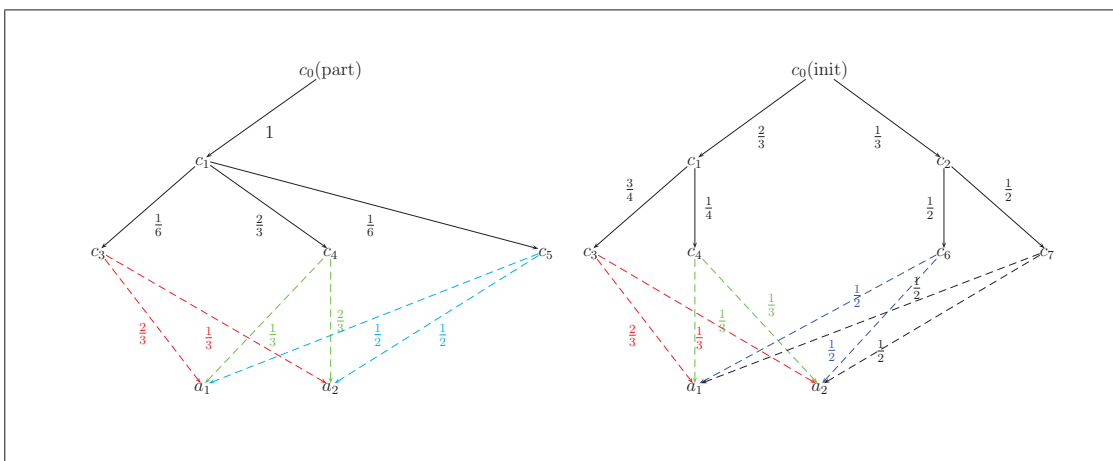


Fig. 11.7.2.I: Hiérarchie décisionnelle évaluée de l'initiateur et du partenaire

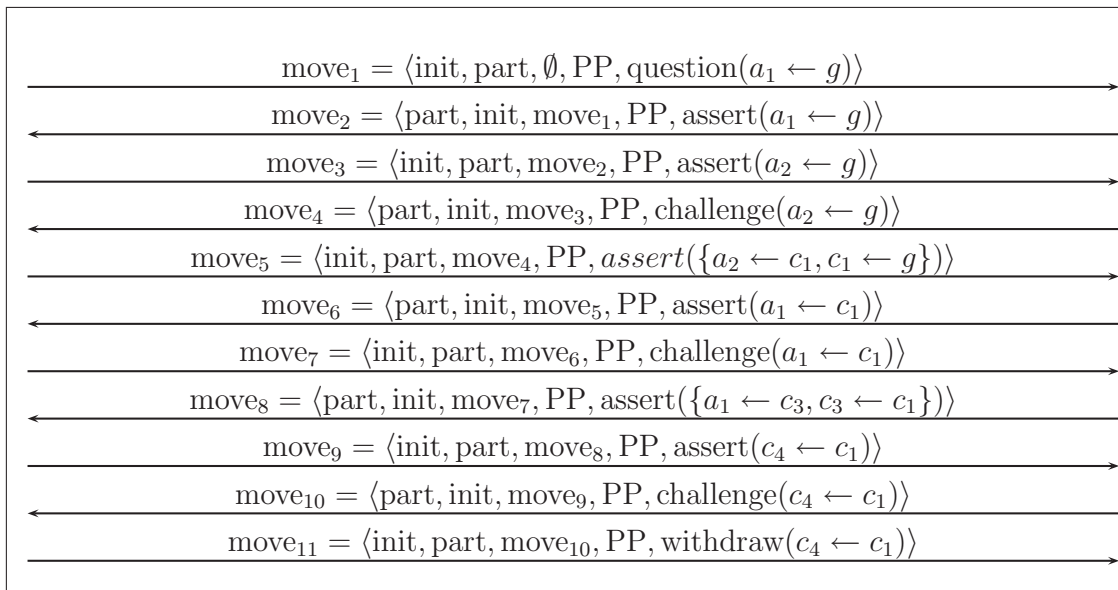


Fig. 11.7.2.II: Persuasion entre deux agents argumentatifs décisionnels

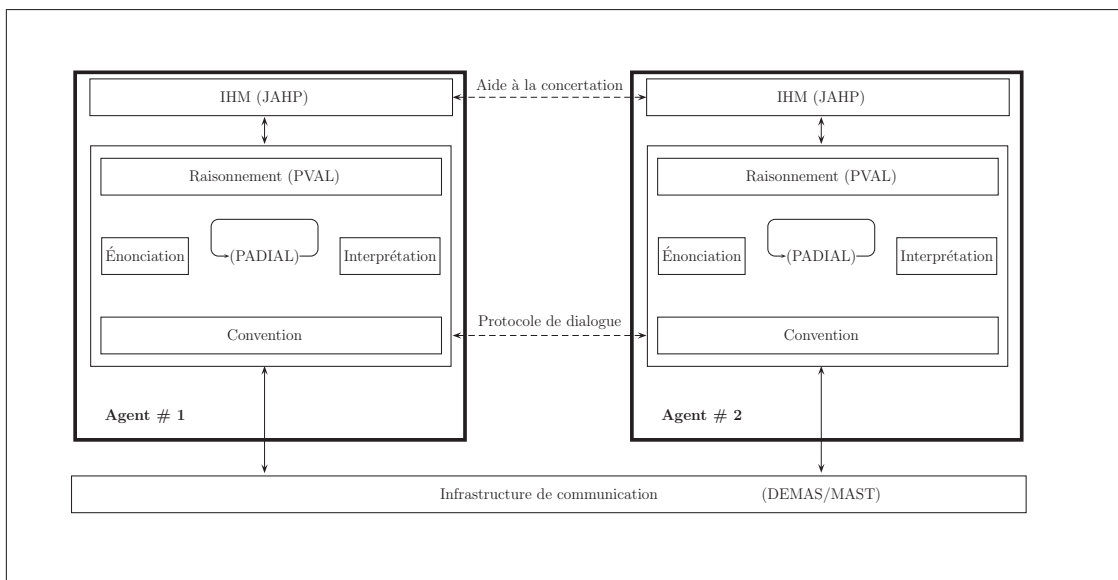


Fig. 11.8.1.I: Architecture du système d'aide à la concertation

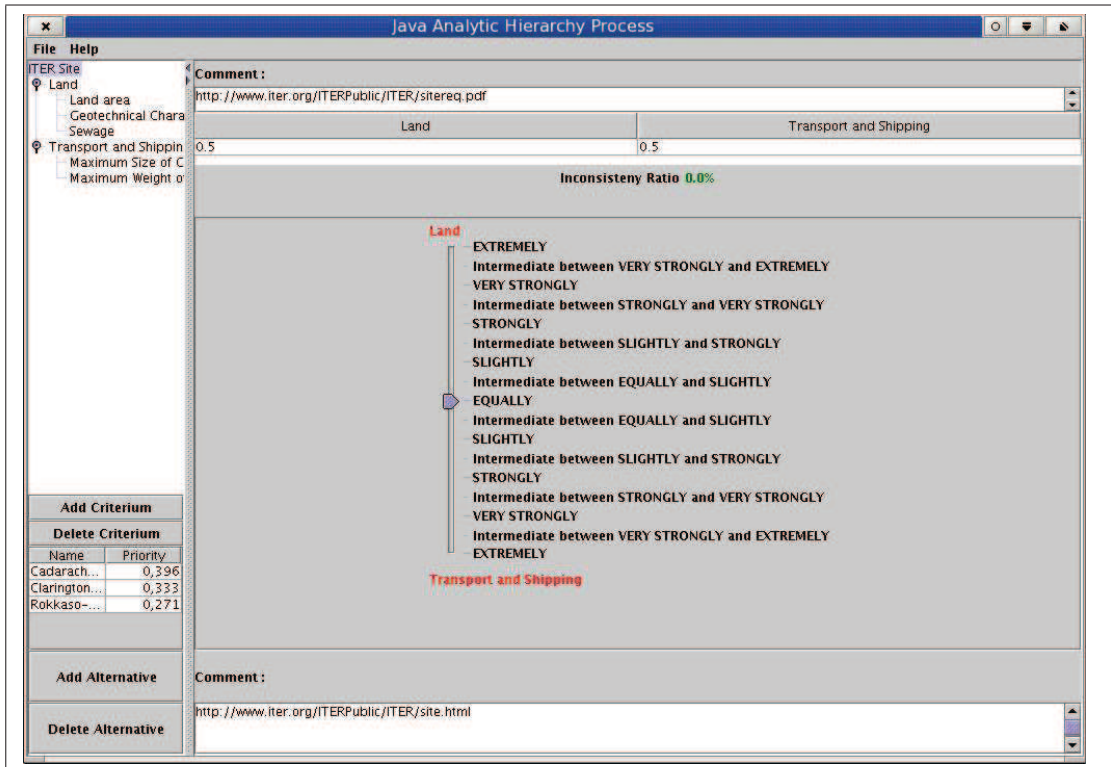


Fig. 11.8.2.I: Capture d'écran de l'interface graphique proposée par JAHP

```

%% Règles
% v1 : objectif
rule(r11, pres(jack), [neg(pres(jose))],v1).%exclusion mutuelle 1
rule(r21, pres(jose), [neg(pres(jack))],v1).%exclusion mutuelle 2
% v2 : sens commun
rule(r12(X),pris(X),[escroc(X)],v2).           %tous les voleurs vont en prison
rule(r22(X),neg(pres(X)),[pris(X)],v2).        %le président ne peut être en prison
% v6 : opinion la + prioritaire
rule(r6,actualpres(jack), [],v6).              %jack est actuellement en prison.
% v5 : opinion
rule(r5,pris(jose), [],v5).                    %jose est en prison.
% v4 : opinion
rule(r4,escroc(jack), [],v4).                   %jack est un escroc
% v3 : opinion la - prioritaire
rule(r3(X),pres(X),[actualpres(X)],v3).        %le président actuel doit le rester
%%Relations de priorité
priority(v1,v2).
priority(v2,v6).
priority(v6,v5).
priority(v5,v4).
priority(v4,v3).

```

Fig. 11.8.3.I: Théorie argumentative mono-agent

```

%%% Tactiques et règle de réponse à une question
%% Attitude coopérative
%si ag est coopératif
assert(SPEAKER,HEARER,CONTENT):-benevolent(ag),
question(HEARER,SPEAKER,CONTENT),willassert(SPEAKER,CONTENT).
%si ag est coopératif
assert(SPEAKER,HEARER,not(CONTENT)):-benevolent(ag),
question(HEARER,SPEAKER,not(CONTENT)),willassert(SPEAKER,not(CONTENT)).
%si ag est égoïste alors aveu d'ignorance
assert(SPEAKER,HEARER,not(CONTENT)):-not(benevolent(ag)),
question(HEARER,SPEAKER,CONTENT).
%% Attitude positive
%si ag est positif
willassert(SPEAKER,CONTENT):-positive(ag),
wantassert(SPEAKER,CONTENT).
%si ag est négatif
willassert(SPEAKER,not(CONTENT)):-not(positive(ag)),
wantassert(SPEAKER,not(CONTENT)).
%% Tactiques d'énonciation d'une affirmation
%si ag est prévenant alors affirme la conclusion d'un argument acceptable
wantassert(_,CONTENT):-thoughtful(ag),convinced(CONTENT).
%si ag est confiant alors affirme la conclusion d'un argument
wantassert(_,CONTENT):-not(thoughtful(ag)),supported(CONTENT).

```

Fig. 11.8.4.I: Règles de réponses à une question

Quatrième partie

Conclusions & Perspectives

Chapitre 12

Conclusion

« Je suis parti d'une croyance plus ou moins religieuse en un monde dans lequel les mathématiques brillaient. Or j'en suis venu à la conclusion que le monde réel est une futilité, et que les mathématiques sont seulement l'art de dire la même chose en des mots différents. »

Bertrand Russel.

Sommaire

12.1 Problématique	213
12.2 Contributions	214
12.3 Limites	215
12.4 Perspectives	215

Dans ce chapitre, nous présentons les conclusions de cette thèse. Nous commençons par rappeler la problématique générale. À la suite de l'exposé de nos contributions, nous soulignons les limites de ce travail de recherche. Enfin, nous en dressons les perspectives à long terme.

12.1 Problématique

Dans cette thèse, nous nous sommes intéressés à la démocratie dialogique. Contrairement à la démocratie représentative qui consiste en un processus d'agrégation des préférences individuelles, la démocratie dialogique est un processus participatif de composition des perspectives et des intérêts. Ce changement de perspective permet, d'un point de vue social et politique, de tirer parti de l'élargissement du cercle de concertation afin d'en enrichir l'expertise et de prévenir d'éventuels blocages. Du point de vue de la prise de décision rationnelle, ce changement de perspective permet d'exploiter la créativité des différents acteurs au cours de la phase de formulation du problème.

Ce travail de recherche a été consacré à la formalisation d'un tel processus à l'aide d'un système multi-agents constitué d'agents cognitifs, autonomes et sociaux en interaction. S'appuyant sur la communication par envoi de messages, on parle de

modèle d'interaction directe entre agents. Ce système multi-agents doit servir de base à un outil informatique d'aide à la concertation.

12.2 Contributions

Dans ce manuscrit, nous avons défendu la thèse selon laquelle le dialogue peut être considéré comme un processus argumentatif au travers duquel les participants jouent et arbitrent pour aboutir à une décision collective. Nous avons proposé ici un **modèle de dialogue entre agents** qui contribue à la formalisation d'une prise de décision concertée. Ce modèle, intitulé **DIAL** (*DIALOGue Is an Argumentative Labour*), sert de base à un **système d'aide à la concertation**.

Nous avons proposé une **logique argumentative à base de valeurs** qui permet de gérer les interactions entre des arguments contradictoires qui sont plus ou moins forts selon l'agent envisagé. Le langage logique sous-jacent permet d'exprimer des connaissances représentant l'état du monde. Plusieurs relations de priorité permettent de considérer simultanément les différentes échelles de valeurs auxquelles se réfèrent les agents. Cette logique argumentative constitue le modèle de raisonnement des agents du modèle DIAL.

Nous avons proposé un **modèle d'agent argumentatif**. Pour que les agents du système échangent leurs connaissances et raisonnent conjointement, ces derniers justifient les hypothèses sur lesquelles ils s'engagent et prennent en compte les engagements de leurs interlocuteurs en fonction de la compétence estimée de l'agent qui a transmis cette information. Un système multi-agents argumentatifs est constitué de tels agents.

Nous avons proposé un **système multi-agents dialectique**. Dans ce cadre formel, un agent tiers, responsable de la décision finale, rend son arbitrage non seulement en fonction des arguments avancés mais également en fonction de son estimation de la compétence relative de chacun des deux joueurs. Le cadre ainsi circonscrit permet de garantir l'obtention d'un résultat au terme du dialogue et d'évaluer sa qualité.

S'appuyant sur le modèle DIAL, nous avons proposé un **système d'aide à la concertation**. Cet outil informatique fournit des fonctionnalités pour l'élaboration collaborative des schémas argumentatifs et pour élucider les consistances et les inconsistances entre les préférences des acteurs et donc détecter les consensus et les conflits.

12.3 Limites

Ayant dessiné les contours de notre contribution, il convient de souligner les limites de ce travail.

La complexité algorithmique du système d'argumentation et des dialogues n'a pas été étudiée avec précision. Toutefois, les résultats préliminaires proposés dans [46] et présentés dans les sections 2.3.1 et 2.4.2 laissent présager l'intractabilité des problèmes de décision envisagés même si nous nous sommes restreints aux fragments des clauses de Horn de la logique du premier ordre. Considérant l'argumentation comme un modèle de raisonnement générique, des instanciations de ce modèle, comme celle que nous avons proposée *via* la méthode hiérarchique multicritère, rendent possible la mise en œuvre de ce modèle de dialogue.

Le tableau 12.3.0.I présente la grille d'analyse des dialogues proposée par Walton et Krabbe [67] (cf section 3.3). Les cinq catégories de dialogues représentées se distinguent de par leur situation initiale et leur but principal. Nous avons formalisé un protocole à réponses uniques pour chacun des dialogues représentés en vert. Certaines hypothèses sur la situation initiale sont nécessaires pour garantir la correction et la complétude de tels dialogues. Il reste à fournir un protocole qui permette de garantir la complétude des enquêtes. De cette manière, aucun argument du système ne pourra être passé sous silence, qu'il soit ou non en possession de l'un des joueurs.

Situation initiale → But principal ↓	Conflit	Problème ouvert	Ignorance d'un participant
Accord stable i.e. Résolution	persuasion	enquête	demande d'information
Règlement pratique i.e. Décision (ou non) d'agir	négociation	délibération	∅

Tab. 12.3.0.I: Aperçu systémique des types de dialogues qui ont (ou non) été abordés

12.4 Perspectives

À plus long terme, plusieurs pistes de recherches sont envisageables pour prolonger (dans la continuité) ceux présentés ici.

Les buts des processus dialectiques envisagés sont uniquement discursifs. Les dialogues portent sur des formules qui représentent un état du monde et non pas des décisions qui devront être ou non mises en œuvre. Pourtant, il pourrait être intéressant de prendre en considération ce que Walton et Krabbe appellent des accords

pratiques (en anglais, *practical settlement*). L'objectif consiste alors à élargir le cadre pour pouvoir y jouer des négociations ou des délibérations (cf figure 12.3.0.I). À cette intention, le langage logique sous-jacent doit être suffisamment expressif pour représenter non seulement des connaissances mais également des buts, des plans et des actions.

Le système dialectique multi-agents nous a permis d'isoler des dialogues. Il pourrait être intéressant d'envisager un cadre dans lequel de tels jeux sont répétés pour y évaluer différentes stratégies d'agents afin de mettre à jour les valeurs de réputation en fonction du résultat des jeux précédents.

Les processus dialectiques que nous avons envisagés n'impliquent que deux joueurs. Il pourrait être intéressant d'étendre ce cadre dyadique à un cadre multi-parties. D'une part, supprimer une telle restriction permettrait aux agents d'entrer ou de sortir du système au cours des dialogues. D'autre part, subdiviser les dialogues multi-parties entre plusieurs canaux sur la base d'une ontologie autoriserait non seulement des observations non-intrusives mais également des suggestions non-sollicitées comme dans les *newsgroups*.

Le processus dialectique proposé compose les intérêts et les perspectives. Toutefois, on ne peut nier l'importance des apprentissages collectifs dans le débat public. Un citoyen détermine ses préférences après avoir pris connaissance des préférences de ses interlocuteurs. À cette intention, les priorités doivent être dynamiques et déterminées au cours du dialogue.

Cinquième partie
Références et index

Bibliographie

- [1] L. Amgoud and C. Cayrol. On the acceptability of arguments in preference-based argumentation framework. In *Proc. of 14th Conference on Uncertainty in Artificial Intelligence*, pages 1–7, 1998. 33, 39, 44, 48, 116, 117, 124
- [2] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Maths and AI*, 34(1-3) :197–215, 2002. 33, 39, 44, 48, 116, 117
- [3] Leila Amgoud, Nicolas Maudet, and Simon Parsons. Modelling dialogues using argumentation. In *Proc. of the 4th International Conference on Multi-Agent Systems*, pages 31–38, Boston, 2000. IEEE Press. 88
- [4] Leila Amgoud, Nicolas Maudet, and Simon Parsons. An argumentation-based semantics for agent communication languages. In F. van Harmelen, editor, *Proc. of the 15th European Conference on Artificial Intelligence*, pages 38–42. IOS Press, Amsterdam, 2002. 88, 117, 118, 142, 143, 148, 153
- [5] Leila Amgoud and Simon Parsons. Agent dialogues with conflicting preferences. In *Proc. of the International Workshop on Agent Theories, Architectures and Languages*, 2001. 88
- [6] John Austin. *How to do things with words?* Harvard Press University, 1962. 10, 11, 12
- [7] FIPA TC B. Fipa acl message representation in xml specification. Component, Foundation for Intelligent Physical Agents, 6-12 2002. <http://fipa.org/specs/fipa00071/>. 15
- [8] FIPA TC B. Fipa acl message structure specification. Component, Foundation for Intelligent Physical Agents, 6-12 2002. <http://fipa.org/specs/fipa00061/>. 16
- [9] T.J.M Bench-Capon. Persuasion in practical argument using value based argument frameworks. *Journal of Logic and Computation*, 13(3) :429–448, 2003. 33, 41, 44, 116, 125, 134

- [10] Jamal Bentahar, Bernard Moulin, and Brahim Chaib-draa. Towards a formal framework for conversational agents. In *Proc. of the workshop on "Agent Communication Languages and Conversation Policies" in AAMAS*, juillet 2003. 108, 109, 111
- [11] Philippe Breiter. *La communication orale coopérative : contribution à la modélisation et à la mise en œuvre d'un agent rationnel dialoguant*. PhD thesis, Université de Paris Nord, 1992. 17, 18
- [12] FIPA TC C. Fipa acl communicative act library specification. Component, Foundation for Intelligent Physical Agents, 6-12 2002. <http://fipa.org/specs/fipa00037/>. 16
- [13] FIPA TC C. Fipa contract net interaction protocol specification. Component, Foundation for Intelligent Physical Agents, 12-03 2003. <http://fipa.org/specs/fipa00029/>. 16, 20
- [14] FIPA TC C. Fipa interaction protocol library specification. Component, Foundation for Intelligent Physical Agents, 2-10 2003. <http://fipa.org/specs/fipa00025/>. 19, 20
- [15] Perelman C. and Olbrechts-Tyteca L. *Traité de l'Argumentation - La Nouvelle Rhétorique*. Presses Universitaires de France, 1958. 32, 57, 125, 141
- [16] C. Castelfranchi and R. Falcone. Principles of trust in mas : Cognitive anatomy, social importance, and quantification. In *Proceedings of ICMAS'98*, pages 72–79, 1998. 141
- [17] Brahim Chaib-draa and Frank Dignum. Trends in agent communication language. *Computational Intelligence*, 18(2) :89–101, 2002. 18, 85
- [18] Gaëlle Desbordes. Un cadre d'interactions d'agents logiciels basé sur l'argumentation. Master of science, Faculté des études supérieures de l'Université Laval, 2001. sous la direction de Bernard Moulin. 32
- [19] Frank Dignum, Barbara Dunin-Keplicz, and Rineke Verbrugge. Dialogue in team formation. In *Issues in Agent Communication*, pages 264–280, London, UK, 2000. Springer-Verlag. 82, 83
- [20] Frank Dignum, Barbara Dunin-Keplicz, and Rineke Verbrugge. Agent theory for team formation by dialogue. In Cristiano Castelfranchi and Yves Lesperance, editors, *Agent Theories Architectures and Languages*, number LNAI 1986 in Intelligent Agents VII, pages 150–166. Springer-Verlag, 2001. 20, 82, 83, 84
- [21] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2) :321–357, 1995. 33, 34, 38, 44, 55, 116

- [22] Peynam Faratin. *Automated Service Negotiation Between Autonomous Computational Agents*. PhD thesis, University of London, Queen Mary College, Department of Electronic Engineering, 2000. 24
- [23] T. F. Gordon and N. Karacapilidis. The Zeno argumentation framework. In *Proceedings of the Sixth International Conference on AI and Law*, pages 10–18, New York, NY, USA, 1997. ACM Press. 189, 199
- [24] Thomas F. Gordon. Computational dialectics. *Computers as Assistants - A New Generation of Support Systems*, pages 186–203, 1996. 186
- [25] Thomas F. Gordon and Oliver Marker. Mediation systems. In Lorenz M. Hilty and Paul W. Gilgen, editors, *15th International Symposium Informatics for Environment Protection*, pages 737–742. Metropolis Verlag, 2001. 186
- [26] Charles L. Hamblin. *Fallacies*. Methuen, 1970. 60, 61, 62
- [27] Andreas Herzig and Dominique Longin. Belief dynamics in cooperative dialogues. *Journal of Semantics*, 17(2) :91–118, may 2000. 146
- [28] Jaako Hintikka. Language-games for quantifiers. *Studies in Logical Theory*, ed. N. Rescher, Basil Blackwell, Oxford, American Philosophical Quarterly Monograph Series, 2 :46–72, 1968. 65
- [29] Marc-Philippe Huget, James Odell, Øystein Haugen, Mariam Misty Nodine, Stephen Cranefield, Renato Levy, and Lin Padgham. Fipa modelling : Interaction diagrams. Technical report, Foundation for Intelligent Physical Agents, 07-02 2003. <http://www.auml.org/auml/documents/ID-03-07-02.pdf>. 19
- [30] Takayuki Ito and Toramatsu Shintani. Persuasion among agents : An approach to implementing a group decision support system based on multi-agent negotiation. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI'97)*. Morgan Kaufmann, 1997. 188
- [31] Mackenzie J.D. Question-begging in non-cumulative systems. *Journal of Philosophical Logic*, 1979. 62
- [32] Antonis C. Kakas and Pavlos Moraitis. Argumentative agent deliberation, roles and context. In Jürgen Dix, João Alexandre Leite, and Ken Satoh, editors, *Electronic Notes in Theoretical Computer Science*, volume 70. Elsevier, 2002. 44, 52, 116, 117, 124, 134
- [33] Nikos I. Karacapilidis and Dimitris Papadias. Computer supported argumentation and collaborative decision making. *Information Systems*, 26(4) :259–277, 2001. 199
- [34] Arrow K.J. *Social choice and individual values*. Wiley, New York, 1963. 187, 188

- [35] Jean Luc Koning. Algorithms for translating interaction protocols into a formal description. In *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 810–815, Tokyo, Japan, October 1999. IEEE Press. 25
- [36] Jean-Luc Koning, G. Francois, and Yves Demazeau. Formalization and pre-validation for interaction protocols in a multi agent systems. In Henri Prade, editor, *Proc. of the 13th European Conference on Artificial Intelligence*, pages 298–307, Brighton, UK, August 1998. John Wiley and Sons. 25
- [37] Marc-André Labrie, Brahim Chaib-draa, and Nicolas Maudet. Diagal : A tool for analyzing and modelling commitment-based dialogues between agents. In Y. Xiang and B. Chaib-draa, editors, *Proc. of the 16th Canadian Conference on Artificial Intelligence*, volume 2671 of *LNAI*, pages 353–369, Halifax, juin 2003. Springer-Verlag. 85, 117
- [38] Paul Lorenzen and Kuno Lorenz. *Dialogische Logik*. Darmstadt : Wissenschaftliche : Buchgesellschaft, 1978. 65
- [39] Nicolas Maudet. *Modéliser les conventions des interactions langagières : la contribution des jeux de dialogues*. PhD thesis, Université Paul Sabatier, Toulouse, Mai 2001. 62, 85
- [40] Nicolas Maudet and Brahim Chaib-draa. Commitment-based and dialogue-game based protocols—news trends in agent communication language. *Knowledge Engineering*, 17(2) :157–179, 2002. 85
- [41] P. McBurney and S. Parsons. Intelligent systems to support deliberative democracy in environmental regulation. *Information and Communications Technology Law*, 10(1) :33–43, 2001. 186
- [42] Pavlos Moraitis. Un modèle de raisonnement pour agents autonomes fondé sur l’argumentation. In *Proc. Journées Nationales de Modèles de Raisonnement (JNMR’03)*, Paris, France, 2003. poster. 44, 52, 117, 124, 134
- [43] John von Neumann Oskar Morgenstern. *The Theory of Games and Economic Behavior*. Princeton University Press, 1944. 186
- [44] Christos H. Papadimitriou. *Computational complexity*. Addison-Welsey Publishing Compagny, 1993. 2nd edition. 243, 244, 245
- [45] Simon Parsons, Carles Sierra, and Nicholas R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 3(8) :261–292, 1998. 81
- [46] Simon Parsons, Michael Wooldridge, and Leila Amgoud. An analysis of formal inter-agent dialogues. In Cristiano Castelfranchi and W. Lewis Johnson, editors, *Proc. of the first international joint conf. on autonomous agent and multiagent*

- systems*, volume part 1, pages 394–401. ACM press, 2002. 38, 88, 142, 143, 148, 153, 215
- [47] Philippe Pasquier, Mathieu Bergeron, and Brahim Chaib-draa. Diagal : a generic acl for open systems. In Marie-Pierre Gleizes et al., editor, *Proc. of the 5th International workshop of Engineering Societies in the Agent World (ESAW'04)*, pages 117–122, Octobre 2004. 85, 117
- [48] Philippe Pasquier and Brahim Chaib-draa. Engagements, intentions et jeux de dialogue. In Andreas Herzig, Brahim Chaib-draa, and Philippe Mathieu, editors, *Actes des Secondes Journées Francophones Modèles Formels de l'Interaction*, pages 289–294, 111 rue Nicolas-Vauquelin, 31100 Toulouse, France, 2003. Cépaduès-Éditions. 20–22 mai 2003, Lille, France. 85
- [49] Henry Prakken. On dialogue systems with speech acts, arguments, and counterrarguments. In M. Ojeda-Aciego, I.P.d. Guzman, G. Brewka, and L.M. Pereira, editors, *Proc. of the 7th European Workshop on Logic for Artificial Intelligence (JELIA'2000)*, number 1919 in Lecture Notes in AI, pages 224–238. Springer Verlag, 2000. 91, 117, 118, 158, 178
- [50] Henry Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *special issue on New Perspectives in Dialogical Logic*, Synthèse 127 :187–219, 2001. 72, 91, 107, 158, 178
- [51] Michael Schroeder Ralf Schweimeier. Notions of attack and justified arguments for extended logic programs. In F. van Harmelen, editor, *Proc. of the 15th European Conference on Artificial Intelligence (ECAI02)*, pages 536–540, Amsterdam, 2002. IOS Press. 44, 45, 47, 51, 116
- [52] Chris A. Reed. Dialogue frames in agent communication. In *Proc. of the 3rd International Conference on Multi Agent Systems (ICMAS)*, pages 246–253, Paris, 1998. IEEE Press. 81
- [53] H. W. J. Rittel and M. M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4 :155–169, 1973. 189
- [54] Thomas L. Saaty. *Decision Making for Leaders ; the Analytical Hierarchy Process for Decisions in a Complex World*. RWS Publications, Pittsburg, 1982. revised 2000. 189
- [55] Thomas L. Saaty. *The Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process*, volume Vol VI of *AHP Series*. RWS Publication, Pittsburg, 1996. revised 2000. 189
- [56] Tuomas Sandholm. *Multiagent Systems*, chapter Distributed Rational Decision Making. The MIT Press, 1999. 186, 187
- [57] Toulmin S.E. *The Uses of Argument*. Cambridge University Press, 1958. 31, 32, 57

- [58] J.R. Searle. *Speech Acts : An Essay in the Philosophy of Language*. Cambridge University Press, 1969. 10, 12
- [59] Amal El Fallah Seghrouchni. *Coordination d'agents : Modèles, Algorithmes et Protocoles*. Memoire d'habilitation a diriger des recherches, Université Paris 13, 2000. 25
- [60] Claude Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 1948. 9
- [61] Reid G. Smith. the contract net protocol : High-level communication and control in a distributed problem solver. *IEEE Transactions on computers*, 29(12) :1104–1113, December 1980. 20, 82
- [62] Aho A. V. and Ullman J. D. *The Theory of Parsing, Translation, and Compiling*, volume 1. Prentice-Hall, 1972. 22
- [63] Daniel Vanderveken. *Analyse et Simulation de Conversations. De la théorie des actes du discours aux systèmes multiagents*, chapter 2 : La structure logique des dialogues intelligents, pages p61–100. L'Interdisciplinaire, 1999. 10
- [64] Daniel Vanderveken and John R. Searle. *Foundations of Illocutionary Logic*. Cambridge University Press, 1985. 10
- [65] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944. 70, 99
- [66] Gerard Vreeswijk and Henry Prakken. Credulous and sceptical argument games for preferred semantics. In M. Ojeda-Aciego, I.P.d. Guzman, G. Brewka, and L.M. Pereira, editors, *Proc. of the 7th European Workshop on Logic for Artificial Intelligence (JELIA '2000)*, number 1919 in Lecture Notes in AI, pages 239–253. Springer Verlag, 2000. 72
- [67] D. Walton and E. Krabbe. *Commitment in Dialogue*. SUNY Press, 1995. 62, 64, 171, 215
- [68] L. Wittgenstein. *Tractatus Logico-Philosophicus*. Routledge & K. Paul, London, 1933. 11

Index

- Cet index référence l'ensemble des notions formalisées dans ce manuscrit, c'est-à-dire l'ensemble des définitions.
- A**
- acceptable 35, 40, 43
 - indéfendable 43
 - objectivement 43, 132
 - selon un agent argumentatif 144
 - subjectivement 43, 132
 - admissible 35, 40, 43, 55
 - Agent argumentatif 89, 140
 - Agent assistant 197
 - Argument 46, 49, 128
 - Argumentaire multi-agents 128
 - Attitude
 - argumentatif 167
 - confiant 150
 - coopérative 165
 - ouverture d'esprit 167
 - prévenant 150
 - égoïste 165
 - Automate
 - de comportement 23
 - fini 22
 - déterministe 22
- B**
- Base d'arguments 100
- C**
- Clôture d'un ensemble d'arguments .. 99
 - Cohérence moyenne 192
 - Complétude 100, 162
 - Compétence estimée 147
- Condition d'énonciation 149
 - d'une acceptation 90
 - d'une affirmation 90, 148
 - d'une concession 148
 - Confrontation 75
 - Consensus 161
 - Convention 74
 - Correction 100, 162
 - Coup 156
 - bien formé 163
 - en discussion 107
 - pertinent 107
 - Crédibilité 160
 - Crédulité 37, 55, 151
- E**
- Engagement social 85, 110
 - règle de mise à jour 90, 93, 146
 - Extension
 - préférée 36
 - raisonnable 35
 - stable 37
- F**
- Fonction caractéristique 35
- H**
- Hierarchie décisionnelle 189
 - valuée 193
- I**
- Indice de cohérence 192
- L**
- Langage

- reconnu 22
- libre de conflit 35, 40, 43
- Ligne persuasive 74, 96, 163
- M**
- Matrice de comparaisons binaires ... 191
- Message 138
- Modèle voir Satisfiabilité
- N**
- Niveau de priorité voir Prioriser
- P**
- Programme
- de logique étendue 45
 - logique 124
 - logique sans négation par l'échec . 52
- Protocole
- d'interaction 18
 - borné 23
 - vivace 23
 - de dialogue
 - de demande d'information 172
 - de persuasion 175
 - à réponses multiples 164
 - à réponses uniques 164
 - de discussion 95
 - bavard 103
 - libertaire 106
 - naïf 101
 - à réponses multiples 97
 - à réponses uniques 97
- Prouvabilité
- crédule 76, 98
 - sceptique 76
- R**
- Ratio de cohérence 192
- Relation
- Attaquer 46, 55, 130
 - vigoureusement 46
 - de contradiction 46
 - Défaire 40, 46
 - Défaire pour un agent 130
 - Prioriser 54, 125
 - Préférer 51
 - Saper 46
 - vigoureusement 46
- Règle de séquence 164
- règle d’Affirmation/Accueil 166
 - règle de Mise en doute/Réponse . 170
 - règle de Question/Réponse 165
 - règle de terminaison 170
- Règle dialectique 61
- S**
- Satisfiabilité 71
- Scepticisme 37, 55, 151
- Stratégie 69, 75, 98
 - gagnante 70, 76, 98
- Système d’argumentation 34
 - cohérent 37
 - spécifique à une audience 42
 - à base de préférences 39, 51
 - à base de valeurs 41
- Système dialectique 68, 73
 - multi-agents 94, 157
- T**
- Tactique
- d'énonciation 151
 - d'une affirmation 150
 - d'une concession 151
- de réponse
 - à une affirmation 167
 - à une question 165
- Théorie argumentative 54
 - décisionnelle 196
 - mono-agent 126
 - multi-agents 125
 - étendue 143

Sixième partie

Annexes

« Les deux mots les plus brefs
et les plus anciens, oui et non,
sont ceux qui exigent le plus de
réflexion. »

Pythagore

Annexe A

Logique classique

Sommaire

A.1 Introduction	229
A.2 Syntaxe	230
A.2.1 Langage propositionnel	230
A.2.2 Langage du premier ordre	230
A.3 Sémantique	232
A.3.1 Logique propositionnelle	233
A.3.2 Logique du premier ordre	234
A.4 Système de preuve	236
A.4.1 Logique propositionnelle	237
A.4.2 Logique du premier ordre	237
A.5 Synthèse	237

A.1 Introduction

Un langage logique est défini par une syntaxe, c'est-à-dire un système de symboles ainsi que des règles pour les combiner dans des formules. De plus, une sémantique est associée au langage. Elle permet de l'interpréter, c'est-à-dire d'attacher à ces formules ainsi qu'aux symboles une signification. Un système de preuve nous permet également de calculer les significations des formules en construisant des démonstrations.

Dans ce chapitre, nous commencerons par définir les symboles et les règles de formation des formules (cf section A.2), que cela soit des formules de la logique des propositions (cf section A.2.1) ou des formules de la logique du premier ordre (cf section A.2.2). Nous aborderons ensuite la sémantique de ces logiques (cf section

A.3). Finalement, nous présenterons brièvement dans la section A.4 un système de preuve pour chacune de ces logiques.

A.2 Syntaxe

Considérons un langage logique. Ce dernier est : soit un langage propositionnel, on parle alors de logique des propositions ; soit un langage du premier ordre, on parle alors de logique des prédicats. Bien évidemment, ces langages logiques diffèrent de part leur syntaxe. Considérons leur syntaxe respective.

A.2.1 Langage propositionnel

La syntaxe de la logique des propositions est fondée sur des variables de propositions appelées également atomes que nous notons avec des lettres minuscules ($p, q, r, s, \text{etc} \dots$). Ces symboles représentent des propriétés qui sont, soit vraies, soit fausses. Ces variables sont combinées au moyen de connecteurs logiques qui sont : le connecteur binaire disjonctif (\vee), le connecteur binaire conjonctif (\wedge), le connecteur binaire de l'implication (\rightarrow), ainsi que le connecteur monadique de la négation (\neg). Ces variables forment alors des formules appelées également propositions. Nous les notons par des lettres grecques minuscules ($\phi, \psi, \theta, \text{etc} \dots$).

Définition n° A.2.1.I [Expression Bien Formée (EBF)]

Soit \mathcal{X} un ensemble de variables de proposition. L'ensemble des **Expressions Bien Formées (EBF)** est défini tel que :

- les symboles propositionnels ($p, q, r, s, \text{etc} \dots$) sont des EBF ;
- si ϕ est une EBF alors $\neg\phi$ est une EBF ;
- si ϕ et ψ sont des EBF alors $\phi \wedge \psi$, $\phi \vee \psi$ ainsi que $\phi \rightarrow \psi$ sont des EBF ;

Afin d'éviter des ambiguïtés possibles, nous utiliserons des parenthèses chaque fois que c'est nécessaire.

Exemple n° A.2.1.I [Expression Bien Formée (EBF)] $(p \vee q) \rightarrow (p \wedge q)$ est une EBF contrairement à $\neg p \rightarrow$.

A.2.2 Langage du premier ordre

La syntaxe de la logique d'ordre un, contrairement à celle d'ordre zéro, considère d'une part les termes qui représentent les objets étudiés, et d'autre part les formules qui sont des propriétés sur ces objets. Dans la suite de ce manuscrit, nous

noterons \mathcal{V} , l'ensemble des variables (x, y, z, \dots) , \mathcal{F} l'ensemble des symboles de fonctions (f, g, \dots) et \mathcal{P} l'ensemble des symboles de prédicats (P, Q, \dots) . On dispose également d'une application dite d'arité, $m : \mathcal{F} \cup \mathcal{P} \rightarrow \mathbb{N}$.

Définition n° A.2.2.I [Termes]

L'ensemble des **termes** \mathcal{T} est défini tel que :

- $x \in \mathcal{T}$ si $x \in \mathcal{V}$;
- $f(t_1, \dots, t_n) \in \mathcal{T}$ si $\forall 1 \leq i \leq n \ t_i \in \mathcal{T}$ et $f \in \mathcal{F}$ avec $m(f) = n$.

On appelle formule atomique ou atome toute application $P(t_1, \dots, t_m)$ d'un symbole de prédicat P d'arité m à m termes.

Comme nous le verrons par la suite, les termes représentent des valeurs dans un domaine donné d'objets, les symboles de fonctions représentent des opérations de base sur ces valeurs et les formules atomiques représentent des propriétés sur ces valeurs. Les formules sont combinées entre elles à l'aide des connecteurs propositionnels usuels, du quantificateur universel $(\forall x)$ et du quantificateur existentiel $(\exists x)$.

Définition n° A.2.2.II [Expression Bien Formée (EBF)]

L'ensemble des **Expressions Bien Formées (EBF)** est défini tel que :

- les formules atomiques sont des EBF;
- si ϕ est une EBF alors $\neg\phi$ est une EBF;
- si ϕ et ψ sont des EBF alors $\phi \wedge \psi$, $\phi \vee \psi$ ainsi que $\phi \rightarrow \psi$ sont des EBF;
- si ϕ est une EBF et $x \in \mathcal{V}$ alors $\exists x\phi$ est une EBF dans la mesure où aucun des quantificateurs $\exists x$ et $\forall x$ n'apparaît dans ϕ ;
- si ϕ est une EBF et $x \in \mathcal{V}$ alors $\forall x\phi$ est une EBF dans la mesure où aucun des quantificateurs $\exists x$ et $\forall x$ n'apparaît dans ϕ .

Nous utiliserons ici encore des parenthèses afin d'éviter des ambiguïtés possibles.

Exemple n° A.2.2.I [Expression Bien Formée (EBF)] Si f est un symbole de fonction avec $m(f) = 1$ et P, Q sont des symboles de prédicats avec $m(P) = 2$ et $m(Q) = 1$ alors : $\forall x (P(f(x), z) \wedge \exists y (P(f(x), y) \vee Q(y)))$ est une EBF contrairement à $\exists x (P(x, y) \wedge \forall x Q(x))$.

Avant d'aborder la sémantique de ce langage, nous devons définir à l'aide de cette syntaxe l'ensemble des variables libres et l'ensemble des variable liées d'une formule.

Définition n° A.2.2.III [Variables libres et liées]

Soit ϕ une formule. L'ensemble des **variables libres** et l'ensemble des **variables liées** de ϕ (notés respectivement $Free(\phi)$ et $Bound(\phi)$) sont définis par récurrence sur la structure de ϕ :

- $Free(x) = \{x\}$ et $Bound(x) = \emptyset$ si $x \in \mathcal{V}$;
- $Free(f(t_1, \dots, t_n)) = Free(t_1) \cup \dots \cup Free(t_n)$ et
 $Bound(f(t_1, \dots, t_n)) = \emptyset$ si $\forall 1 \leq i \leq n \ t_i \in \mathcal{T}$ et $f \in \mathcal{F}$ avec $m(f) = n$;
- $Free(P(t_1, \dots, t_n)) = Free(t_1) \cup \dots \cup Free(t_n)$ et
 $Bound(P(t_1, \dots, t_n)) = \emptyset$ si $\forall 1 \leq i \leq n \ t_i \in \mathcal{T}$ et $P \in \mathcal{P}$ avec $m(P) = n$;
- $Free(\neg\phi) = Free(\phi)$ et $Bound(\neg\phi) = Bound(\phi)$;
- $Free(\phi \wedge \psi) = Free(\phi \vee \psi) = Free(\phi \rightarrow \psi) = Free(\phi) \cup Free(\psi)$
 et $Bound(\phi \wedge \psi) = Bound(\phi \vee \psi) = Bound(\phi \rightarrow \psi) = Bound(\phi) \cup Bound(\psi)$;
- $Free(\forall x\phi) = Free(\exists x\phi) = Free(\phi) - \{x\}$ et $Bound(\forall x\phi) = Bound(\exists x\phi) = Bound(\phi) \cup \{x\}$.

Dans les formules $\forall x\phi$ et $\exists x\phi$, ϕ est appelé la **portée** du quantificateur. En conséquence, une variable est libre si elle n'est dans la portée d'aucun quantificateur. Sinon elle est liée. On appelle alors une **formule fermée** (ou close) une formule qui ne contient pas de variable libre. Sinon elle est **ouverte**.

La notion de substitution est également utile pour la définition de la sémantique.

Définition n° A.2.2.IV [Substitution de variables]

Une substitution de variables est une application de l'ensemble des variables vers l'ensemble des termes. Le résultat d'une substitution de x par t dans ϕ est la formule obtenue quand toutes les occurrences de x ont été remplacées par des occurrences de t . On le note $\phi[t/x]$.

Exemple n° A.2.2.II [Substitution de variables] Le résultat de la substitution $(\forall x\exists yP(x, y))[t/x]$ est le suivant : $\forall t\exists yP(t, y)$.

A.3 Sémantique

Qu'en est-il de la signification d'une formule ? C'est l'objet de la sémantique. Là encore, elle diffère selon le langage envisagé.

A.3.1 Logique propositionnelle

En logique propositionnelle, une formule est soit vraie soit fausse. Plus formellement, l'ensemble des valeurs de vérité est un ensemble \mathcal{B} de deux booléens : le vrai (1 noté également \top) et le faux (0 noté également \perp). La signification des booléens est définie à l'aide de fonctions de booléens vers des booléens. Ces fonctions peuvent être représentées sous la forme de table de vérité (cf figure A.3.1.I).

\wedge	\perp	\top	\vee	\perp	\top	\rightarrow	\perp	\top	\neg	\perp	\top
\perp	\perp	\perp	\perp	\perp	\top	\perp	\top	\top	\perp	\top	\perp
\top	\perp	\top	\top	\top	\top	\top	\perp	\perp	\top	\perp	\top

Fig. A.3.1.I: Tables de vérité des connecteurs logiques

La signification d'une formule dépend donc de la valeur de vérité de ses variables et est définie formellement ci-dessous :

Définition n° A.3.1.I [Interprétation de la logique des propositions]

Soit \mathcal{X} un ensemble de variable. Une **interprétation** I (appelée également *assignation*) est une application de l'ensemble des variables vers \mathcal{B} . La sémantique d'une expression bien formée ϕ dans l'interprétation I (notée $|\phi|_I$) est définie par récurrence sur sa structure :

- $|\phi|_I = I(p)$ avec $\phi = p$ et $p \in \mathcal{X}$;
- $|\neg\phi|_I = 1 - |\phi|_I$;
- $|\phi \wedge \psi|_I = \min\{|\phi|_I, |\psi|_I\}$;
- $|\phi \vee \psi|_I = \max\{|\phi|_I, |\psi|_I\}$;
- $|\phi \rightarrow \psi|_I = \max\{1 - |\phi|_I, |\psi|_I\}$.

On dit d'une formule ϕ qu'elle est vraie dans l'interprétation I ssi $|\phi|_I = 1$. Dans le cas contraire ($|\phi|_I = 0$), elle est fausse.

Exemple n° A.3.1.I [Interprétation de la logique des propositions] Si $|p|_I = 1$ et $|q|_I = 0$ alors $|p \vee q|_I = |p \vee \neg p|_I = 1$ et $|p \wedge q|_I = |p \rightarrow q|_I = |p \wedge \neg p|_I = 0$.

On est alors en mesure de définir la notion de satisfiabilité et de validité d'une formule.

Définition n° A.3.1.II [Modèle]

Soit ϕ une formule et I une interprétation.

- I est un **modèle** de ϕ , ou I satisfait ϕ (noté $I \models \phi$) ssi $|\phi|_I = \top$;
- un ensemble ψ de formules appelé théorie **entraîne** ϕ (noté $\psi \models \phi$) ssi tous les modèles de ψ sont aussi des modèles de ϕ ;
- ϕ est **valide** ssi ϕ est vraie dans toutes affectations (noté $\models \phi$). Dans le cas contraire on dit que ϕ est **invalid**. Une formule propositionnelle valide est aussi appelée une **tautologie** ;
- ϕ est **satisfiable** ssi elle est vraie dans au moins une affectation (*i.e.* ϕ a au moins un modèle). Dans le cas contraire, elle est **insatisfiable**.

Au même titre que toutes les formules valides sont satisfiables, toutes les formules insatisfiables sont invalides. En conséquence, on distingue 3 types de formules : les formules valides *i.e.* celles qui sont toujours vraies, les formules insatisfiables *i.e.* celles qui sont toujours fausses, et les formules à la fois invalides et satisfiables *i.e.* celles qui sont parfois vraies et parfois fausses. De plus, la validité et l'insatisfiabilité se correspondent via la négation. ϕ est valide si et seulement si $\neg\phi$ est insatisfiable. De même ϕ est insatisfiable si et seulement si $\neg\phi$ est valide.

Exemple n° A.3.1.II [Modèle] D'après l'exemple précédent :

- $p \vee \neg p$ est une formule valide ;
- $p \wedge \neg p$ est une formule insatisfiable (on parle du tiers-exclu) ;
- $p \vee q$ et $p \wedge q$ sont invalides et satisfiables.

A.3.2 Logique du premier ordre

Comme dans le cas propositionnel, la sémantique de la logique du premier ordre est décrite par une interprétation. Cependant le langage de la logique du premier ordre est plus riche. En conséquence, de nouvelles définitions sont nécessaires. Contrairement au langage propositionnel, les interprétations et les affectations sont des objets différents. Une affectation donne une valeur à chaque variable, alors qu'une interprétation décrit le domaine des valeurs et la sémantique des symboles de fonctions et de prédicats.

Définition n° A.3.2.I [Interprétation de la logique du premier ordre]

Une **interprétation** du langage d'ordre un est une structure $M = \langle D, I \rangle$ où D est un ensemble non-vide appelé domaine d'interprétation et I est une fonction d'interprétation c'est-à-dire une application de D^n vers D pour chaque symbole de fonction f d'arité n , et une application de D^n vers \mathcal{B} pour chaque symbole de prédicat P d'arité n . Une affectation g est une application de \mathcal{V} vers D . La sémantique d'une expression bien formée ϕ dans l'interprétation I modulo l'affectation g (noté $|\phi|_{Ig}$) est définie par récurrence sur sa structure :

- $|x|_{Ig} = g(x)$ avec $x \in \mathcal{X}$;
- $|f(t_1, \dots, t_n)|_{Ig} = I(f)(|t_1|_{Ig}, \dots, |t_n|_{Ig})$ si $\forall 1 \leq i \leq n$ $t_i \in \mathcal{T}$ et $f \in \mathcal{F}$ avec $m(f) = n$;
- $|P(t_1, \dots, t_n)|_{Ig} = I(P)(|t_1|_{Ig}, \dots, |t_n|_{Ig})$ si $\forall 1 \leq i \leq n$ $t_i \in \mathcal{T}$ et $P \in \mathcal{P}$ avec $m(P) = n$;
- $|\neg\phi|_{Ig} = 1 - |\phi|_{Ig}$;
- $|\phi \wedge \psi|_{Ig} = \min\{|\phi|_{Ig}, |\psi|_{Ig}\}$;
- $|\phi \vee \psi|_{Ig} = \max\{|\phi|_{Ig}, |\psi|_{Ig}\}$;
- $|\phi \rightarrow \psi|_{Ig} = \min\{1 - |\phi|_{Ig}, |\psi|_{Ig}\}$;
- $|\forall x\phi|_{Ig} = \min_{v \in D} |\phi|_{I(g[v/x])}$;
- $|\exists x\phi|_{Ig} = \max_{v \in D} |\phi|_{I(g[v/x])}$;

Le domaine est ce que nous interprétons comme l'objet du langage. C'est dans le domaine que se situent les variables. Une interprétation fournit un ensemble de valeurs D , un ensemble d'opérations $I(f)$ sur ces valeurs, et un ensemble de propriétés de base $I(P)$ sur les n-uplets de valeurs. Une interprétation nous permet d'associer aux termes des valeurs dans D et d'associer aux formules des valeurs de vérité.

On dit d'une formule ϕ qu'elle est vraie dans l'interprétation I modulo l'affectation g ssi $|\phi|_{Ig} = 1$. Dans le cas contraire ($|\phi|_{Ig} = 0$), elle est fausse.

Exemple n° A.3.2.I [Interprétation de la logique du premier ordre] Soit $M = \langle D, I \rangle$ une interprétation définie telle que $D = \{\alpha, \beta\}$. La fonction d'interprétation I est définie sur le symbole de prédicat P telle que $|P(\alpha, \beta)|_{Ig} = 1$ et $|P(\alpha, \alpha)|_{Ig} = |P(\beta, \beta)|_{Ig} = |P(\beta, \alpha)|_{Ig} = 0$. Soit g une fonction d'assignation définie telle que $g(x) = \alpha$ et $g(y) = \beta$. La formule $P(x, y)$ est vraie dans l'interprétation I modulo l'affectation g . Les formules $P(x, x)$, $P(y, y)$ et $P(y, x)$ sont fausses dans l'interprétation I modulo l'affectation g .

On est alors en mesure de définir la notion de satisfiabilité et de validité d'une formule.

Définition n° A.3.2.II [Modèle]

Soit ϕ une formule du premier ordre, et M une interprétation.

- ϕ est **valide** si elle est vraie dans toute interprétation et toute affectation. Dans le cas contraire on dit que ϕ est **invalidé** ;
- ϕ est **insatisfiable** si elle est fausse dans toute interprétation et toute affectation. Dans le cas contraire on dit que ϕ est satisfiable.
- si ϕ est satisfiable dans M avec la fonction d'affectation g alors l'interprétation $\langle M, g \rangle$ est appelée un **modèle** de ϕ (noté $\langle M, g \rangle \models \phi$).

Ainsi, la conséquence sémantique lie une formule ϕ à une autre formule ψ (notée $\phi \models \psi$) si tout modèle de ϕ est aussi un modèle de ψ .

Exemple n° A.3.2.II [Modèle] D'après l'exemple précédent $\langle M, g \rangle$ est un modèle de $P(x, y)$. Cette formule est satisfiable mais invalidé dans M .

A.4 Système de preuve

Nous avons dans la section précédente doté la logique propositionnelle ainsi que la logique du premier ordre d'une sémantique. Toutefois, il est difficile¹ de l'utiliser pour décider si une formule est satisfiable (ou non) voire valide (ou non). Il faudrait pour cela énumérer toutes les interprétations. Leur nombre est exponentiel. Une alternative consiste à examiner les preuves bien formées, et considérer leurs conclusions. Pour cela nous utilisons un système de preuve.

Définition n° A.4.0.III [Système de preuve]

Un **système de preuve** est un couple $(\mathcal{A}, \mathcal{R})$, où \mathcal{A} est un ensemble de formules appelées axiomes et \mathcal{R} est un ensemble de règles d'inférence, c'est-à-dire des relations entre ensembles de formules (les prémisses) et des formules (la conclusion).

On appelle dérivation à partir d'un ensemble d'hypothèses une suite non vide de formules qui sont : soit des axiomes, soit des formules déduites des formules précédentes de la suite. Une preuve d'une formule ϕ à partir d'un ensemble de formule Γ est une dérivation à partir de Γ dont la dernière formule est ϕ . S'il existe une preuve de ϕ à partir de Γ , nous disons que Γ prouve ϕ , ou que ϕ est prouvable à partir de Γ (on note $\Gamma \vdash_{(\mathcal{A}, \mathcal{R})} \phi$). Un théorème est une formule qui est prouvable à partir de l'ensemble vide d'hypothèses.

Bien évidemment, les système de preuve diffèrent selon le langage logique associé.

¹au sens de la complexité algorithmique

A.4.1 Logique propositionnelle

Un exemple classique de système de preuve pour la logique propositionnelle classique est le système \mathcal{SKC} . Dans ce système, \Rightarrow est le seul connecteur logique et F la seule constante. Ses axiomes sont :

$$(S) (\phi \Rightarrow \phi' \Rightarrow \phi'') \Rightarrow (\phi \Rightarrow \phi') \Rightarrow (\phi \Rightarrow \phi'');$$

$$(K) \phi \Rightarrow \phi' \Rightarrow \phi;$$

$$(C) \neg\neg\phi \Rightarrow \phi.$$

Sa seule règle d'inférence est le modus ponens :

$$(MP) \text{ de } \phi \text{ et } \phi \Rightarrow \phi', \text{ on déduit } \phi'.$$

Ce système de preuve est correct : si $\Gamma \vdash_{\mathcal{SKC}} \phi$ alors $\Gamma \models \phi$. Il est complet : si $\Gamma \models \phi$ alors $\Gamma \vdash_{\mathcal{SKC}} \phi$. Il est également décidable : il existe un algorithme qui, étant donné Γ un ensemble fini de formules et une formule ϕ , décide si $\Gamma \vdash_{\mathcal{SKC}} \phi$.

A.4.2 Logique du premier ordre

Les systèmes de preuve pour la logique du premier ordre sont des extensions de ceux pour la logique propositionnelle, où là, les variables propositionnelles sont remplacées par des atomes, et où de nouvelles règles sont ajoutées pour tenir compte des quantifications.

Un exemple classique de système de preuve pour la logique du premier ordre est le système \mathcal{SKC}_1 . Dans ce système, seuls les symboles de base \Rightarrow , F et \forall sont utilisés.

Ses axiomes sont :

$$(S) (\phi \Rightarrow \phi' \Rightarrow \phi'') \Rightarrow (\phi \Rightarrow \phi') \Rightarrow (\phi \Rightarrow \phi'');$$

$$(K) \phi \Rightarrow \phi' \Rightarrow \phi;$$

$$(C) \neg\neg\phi \Rightarrow \phi;$$

$$(E) (\forall x\phi) \Rightarrow \phi[t/x] \text{ avec } t \in \mathcal{T} \text{ et } x \in \mathcal{V};$$

$$(I) (\forall x\phi \Rightarrow \phi') \Rightarrow \phi \Rightarrow \forall x\phi' \text{ avec } x \in \mathcal{V} \cap \text{Bound}(\phi).$$

Les règles d'inférence sont :

$$(MP) \text{ de } \phi \text{ et } \phi \Rightarrow \phi', \text{ on déduit } \phi';$$

$$(Gen) \text{ de } \phi \text{ on déduit } \forall x\phi \text{ avec } x \in \mathcal{V}.$$

Ce système de preuve est correct et complet mais il n'est pas décidable.

A.5 Synthèse

Nous avons présenté dans ce chapitre les éléments essentiels de la **logique classique**, c'est-à-dire de la logique telle qu'elle est communément appréhendée. Primo, un langage logique est muni d'une syntaxe. Les symboles logiques sont combinés selon

certaines règles pour former des formules appelées expressions bien formées. Secundo, la sémantique de la logique permet d'assigner une signification à ces formules. On dit de ces formules qu'elles sont satisfiables ou insatisfiables, valides ou invalides. Tertio, c'est au travers d'un système de preuve, c'est-à-dire un ensemble d'axiomes et de règles d'inférence, que l'on peut calculer la satisfiabilité de ces formules.

Qu'il s'agisse de logique propositionnelle ou de logique des prédicats, l'opération de déduction en logique classique (\vdash) est une opération qui vérifie un certain nombre de propriétés, notamment la monotonie. Selon cette propriété, si Φ est une conséquence de Γ alors c'est également une conséquence de tout ensemble qui contient Γ :

$$\text{si } \Gamma \vdash \Phi \text{ et } \Gamma \subset \Delta \text{ alors } \Delta \vdash \Phi.$$

Comme cela est abordé dans le chapitre 2, une telle hypothèse ne convient pas toujours. Remettre en cause une telle hypothèse consiste à explorer une **logique dite non classique**.

“I conjecture that there is no good algorithm for the traveling salesman problem. My reasons are the same as for any mathematical conjecture : (1) It is a legitimate mathematical possibility, and (2) I do not know.”

Jack Edmonds

Annexe B

Complexité computationnelle

Sommaire

B.1 Introduction	239
B.2 Algorithme	240
B.3 Analyse algorithmique	240
B.4 Problème de décision	242
B.4.1 Réduction algorithmique	243
B.4.2 Complexité en espace	244
B.5 Problème de recherche	245
B.6 Synthèse	247

B.1 Introduction

Tout comme l’homme, la machine calcule ! Si vous avez déjà ouvert un livre de recettes de cuisine ou déchiffré un mode d’emploi traduit directement du coréen pour faire fonctionner un magnétoscope ou un répondeur téléphonique réticents, sans le savoir, vous avez déjà exécuté des algorithmes¹. Les algorithmes sont des outils pour la résolution de problèmes. Évaluer leur efficacité permet de classer ces problèmes selon leur difficulté intrinsèque.

Nous allons dans un premier temps définir précisément ce qu’est un algorithme (cf section B.2). La section B.3 présente les outils mathématiques nécessaires à l’évaluation de leur efficacité. La section B.4 propose une typologie des problèmes de décision qui s’appuie sur la difficulté calculatoire de leur résolution. La section B.5 propose ce même type de typologie pour les problèmes de recherche.

¹ne pas confondre avec l’*agglô rythmique*, qui consiste à poser des parpaings en cadence.

B.2 Algorithme

Un **algorithme** est une procédure de calcul constituée d'une suite d'instructions, qui prend en **entrée** une valeur ou un ensemble de valeurs et qui une fois exécutée correctement, conduit à un résultat donné : une valeur ou un ensemble de valeur appelé **sortie**. Un algorithme est un outil de résolution d'un **problème abstrait** bien défini. Celui-ci doit énoncer la relation désirée entre l'entrée et la sortie en des termes généraux mais non-ambigus. Si l'algorithme est **correct**, la sortie correspond au résultat désiré. On dit alors que l'algorithme **résout** le problème posé. Un **algorithme** décrit une procédure de calcul spécifique pour établir cette relation. Il peut exister plusieurs algorithmes candidats pour un problème donné. Il est alors nécessaire de les évaluer.

B.3 Analyse algorithmique

Analyser un algorithme consiste à prévoir les ressources qui lui seront nécessaires, en temps et en espace, pour résoudre le problème. La **taille de l'entrée** est le nombre d'éléments constituant l'entrée. Elle dépend du problème, par exemple, la somme des tailles des listes dans le cas d'un problème de fusion de deux listes. Dans la plupart des cas le temps d'exécution et l'espace mémoire de stockage d'un algorithme croissent en fonction de la taille de son entrée.

L'**analyse empirique** consiste à étudier la complexité d'un algorithme, en temps et en espace, de manière expérimentale. À cet effet, l'algorithme doit être implémenté et testé sur un ensemble d'entrées de tailles et de compositions différentes. Toutefois cette méthode ne permet de comparer des algorithmes qu'à la condition de disposer d'implémentations réalisées dans un même environnement informatique. D'autre part, les résultats obtenus ne sont pas nécessairement représentatifs de toutes les entrées.

L'**analyse théorique** consiste à évaluer un pseudo-code de l'algorithme. Le **temps d'exécution** d'un algorithme sur une **instance**, *i.e.* une donnée particulière de toutes les données, est le nombre d'opérations élémentaires exécutées, appelées précédemment instructions. L'**espace mémoire** d'un algorithme sur une instance est le nombre d'éléments manipulés en mémoire à un instant donné de la résolution. Le temps d'exécution pour une entrée quelconque est estimé par une borne supérieure : le **temps d'exécution dans le pire des cas**, en d'autres termes le plus long temps d'exécution pour une entrée quelconque. Le temps d'exécution peut parfois être estimé à l'aide du **temps d'exécution moyen** (ou espéré). Il en va de même pour l'espace mémoire.

Dans la grande majorité des cas, seul l'ordre de grandeur nous intéresse.

Lorsque la complexité d'un algorithme, en temps ou en espace, est exprimée sous la forme d'un polynôme $E/T(n) = \sum_{i=0}^p a_i n^i$ où n est la taille de l'entrée et les a_i sont des constantes. Nous négligerons non seulement le coefficient constant du premier terme mais également les termes d'ordre inférieur. On note alors $T/E(n) \in \theta(n^p)$ et par abus de langage $T/E(n) = \theta(n^p)$. Cette notation est explicitée de manière plus formelle dans la définition suivante.

Définition n° B.3.0.I [Borne approchée asymptotique]

On note $\theta(g(n))$ l'ensemble des fonctions $f(n)$ définies telles que :

$$\exists c_1, c_2 \in \mathbb{R}_+^* \exists n_0 \in \mathbb{N}^* \forall n \geq n_0, 0 \leq c_1.g(n) \leq f(n) \leq c_2.g(n) \quad (\text{B.3.0.I})$$

On dit que $g(n)$ est **une borne approchée asymptotique** pour $f(n)$.

Cette notation signifie qu'une fonction est bornée asymptotiquement à la fois par excès et par défaut. La notation qui suit est utilisée lorsqu'on ne dispose que d'une borne inférieure.

Définition n° B.3.0.II [Borne asymptotique inférieure]

On note $\Omega(g(n))$ l'ensemble des fonctions $f(n)$ définies telles que :

$$\exists c \in \mathbb{R}_+^* \exists n_0 \in \mathbb{N}^* \forall n \geq n_0, 0 \leq c.g(n) \leq f(n) \quad (\text{B.3.0.II})$$

On dit que $g(n)$ est **une borne asymptotique inférieure** pour $f(n)$.

À l'inverse, la notation suivante est utilisée lorsqu'on ne dispose que d'une borne supérieure.

Définition n° B.3.0.III [Borne asymptotique supérieure]

On note $\mathcal{O}(g(n))$ l'ensemble des fonctions $f(n)$ définies telles que :

$$\exists c \in \mathbb{R}_+^* \exists n_0 \in \mathbb{N}^* \forall n \geq n_0, 0 \leq f(n) \leq c.g(n) \quad (\text{B.3.0.III})$$

On dit que $g(n)$ est **une borne asymptotique supérieure** pour $f(n)$.

La borne supérieure $g(n)$ fournie peut être ou non asymptotiquement approchée. On utilise la notation suivante pour dire qu'elle ne l'est pas.

Définition n° B.3.0.IV [Borne supérieure non-asymptotique]

On note $o(g(n))$ l'ensemble des fonctions $f(n)$ définies telles que :

$$\forall c \in \mathbb{R}_+^* \exists n_0 \in \mathbb{N}^* \forall n \geq n_0, 0 \leq f(n) \leq c.g(n) \quad (\text{B.3.0.IV})$$

On dit que $g(n)$ est **une borne supérieure non-asymptotique** pour $f(n)$.

Afin d'illustrer la nuance ici introduite, considérons les cas suivants.

Exemple n° B.3.0.I [Efficacité asymptotique] $2n = \mathcal{O}(n^2)$ et $2n^2 = \mathcal{O}(n^2)$ ont ici la même borne asymptotique. De plus, n^2 est approximativement approchée de $2n^2$ (on note $2n^2 \neq o(n^2)$) mais pas de $2n$ (on note $2n = o(n^2)$).

Il est communément admis qu'un algorithme dont le temps d'exécution est en $\mathcal{O}(n^k)$ avec k constante est raisonnable. Mais dispose-t-on toujours d'un tel algorithme ?

B.4 Problème de décision

Un problème abstrait définit la relation entre un ensemble d'entrées et un ensemble de solutions. Nous ne considérons dans un premier temps que les **problèmes de décision**. L'ensemble de définition des solutions est alors $\{0, 1\}$. En d'autres termes, l'ensemble des problèmes de décisions peuvent être formulés par une question totale. Le problème de décision **complémentaire** est celui dont la question est renversée.

On distingue parmi les problèmes de décision deux classes : la classe des problèmes \mathcal{P} (*Polynomial time*) qui sont relativement faciles à résoudre, et la classe des problèmes plus difficiles \mathcal{NP} (*Nondeterministic Polynomial time*).

Définition n° B.4.0.V [Classes des problèmes de décision]

Parmi les problèmes de décision on distingue :

- la **classe** \mathcal{P} des problèmes de décision qui admettent un algorithme polynomial en temps, c'est-à-dire tels que : $\exists k \in \mathbb{N}, T(n) = \mathcal{O}(n^k)$;
- la **classe** $\text{co}\mathcal{P}$ des problèmes de décision dont le problème complémentaire est dans \mathcal{P} ;
- la **classe** \mathcal{NP} des problèmes de décision qui admettent un algorithme non-déterministe polynomial en temps ;
- la **classe** $\text{co}\mathcal{NP}$ des problèmes de décision dont le problème complémentaire est dans \mathcal{NP} .

Contrairement aux algorithmes déterministes qui exécutent une séquence d'instructions déterminées, les algorithmes (ou plutôt machines) non déterministes deviennent

la solution et procèdent de manière déterministe pour la valider. Ces algorithmes ne croient pas en une solution à tort mais doivent toujours vérifier la validité de la solution. Ce concept théorique constitue la base de toute la théorie de la \mathcal{NP} -complétude. En d'autres termes, la classe \mathcal{NP} contient l'ensemble des problèmes qui possèdent un **certificat polynomial**, c'est-à-dire un algorithme polynomial capable de vérifier si une instance du problème est positive.

Il est relativement évident que $\mathcal{P} \subseteq \mathcal{NP}$, $\mathcal{P} = \text{co}\mathcal{P}$ et que $\mathcal{P} \subset \text{co}\mathcal{NP} \cap \mathcal{NP}$. À l'inverse, $\mathcal{P} \neq \mathcal{NP}$, $\mathcal{P} \neq (\mathcal{NP} \cap \text{co}\mathcal{NP})$ et $\text{NP} \neq \text{co}\mathcal{NP}$ ne sont actuellement que des conjectures. Toutefois, l'existence des problèmes \mathcal{NP} -complet nous pousse à croire que $\mathcal{P} \neq \mathcal{NP}$.

La notion de réduction algorithmique présentée dans la section B.4.1 est un argument en faveur de cette conjecture. Nous allons dans la section B.4.2 situer la complexité en temps par rapport à la complexité en espace.

B.4.1 Réduction algorithmique

Un problème de décision A peut être ramené à un autre problème de décision B si n'importe quelle entrée de A peut être facilement reformulée comme une entrée de B .

Définition n° B.4.1.I [Réduction algorithmique]

Soit A et B deux problèmes de décision et x une instance positive de A . Une **réduction algorithmique** de A vers B noté $A \alpha_\epsilon B$ est définie telle que :

$$A \alpha_\epsilon B \Leftrightarrow x \in A \text{ ssi } f(x) \in B \text{ avec } \epsilon \text{ complexité de } f. \quad (\text{B.4.1.I})$$

Les réductions polynomiales permettent de démontrer qu'un problème est au moins aussi difficile qu'un autre.

Théorème n° B.4.1.I [Réduction polynomiale]

$$\text{si } A \alpha_P B \text{ et } B \in \mathcal{P} \text{ alors } A \in \mathcal{P} \quad (\text{B.4.1.II})$$

$$\text{si } A \alpha_P B \text{ et } B \in \mathcal{NP} \text{ alors } A \in \mathcal{NP} \quad (\text{B.4.1.III})$$

Démonstration n° B.4.1.I [Réduction polynomiale] Cette démonstration est disponible dans [44].

Parmi l'ensemble des problèmes appartenant à \mathcal{NP} , il en existe un sous-ensemble qui contient les problèmes les plus difficiles : on les appelle les problèmes \mathcal{NP} -complet.

Définition n° B.4.1.II [Problème \mathcal{NP} -difficile]

Soit A un problème de décision. A est \mathcal{NP} -difficile ssi :

$B \alpha_P A \forall B \in \mathcal{NP}$.

Un problème \mathcal{NP} -complet est un problème \mathcal{NP} -difficile dans \mathcal{NP} .

Définition n° B.4.1.III [Problème \mathcal{NP} -complet]

Soit A un problème de décision. A est \mathcal{NP} -complet ssi :

$$A \in \mathcal{NP} \quad (\text{B.4.1.IV})$$

$$B \alpha_P A \forall B \in \mathcal{NP} \quad (\text{B.4.1.V})$$

Par conséquent, si A et B sont deux problèmes de décision dans \mathcal{NP} et si A est \mathcal{NP} -complet, il suffit de proposer une réduction algorithmique ($A \alpha_P B$) pour prouver que B est \mathcal{NP} -complet. À cette intention, on peut utiliser SAT qui est, d'après le théorème de Cook [44], \mathcal{NP} -complet.

Qu'en est-il de la complexité en espace des algorithmes pour résoudre ces classes de problèmes ? C'est l'objet de la section suivante.

B.4.2 Complexité en espace

Alors les classes des problèmes précédemment définies (\mathcal{P} , \mathcal{NP} , ...) prennent en considération le temps d'exécution des algorithmes, les classes de problèmes suivantes prennent en considération l'espace utilisé.

Définition n° B.4.2.I [Classes des problèmes de décision]

\mathcal{L} -space est la classe des problèmes de décision qui utilisent un espace de travail logarithmique en la taille des données.

\mathcal{P} -space est la classe des problèmes de décision qui utilisent un espace de travail polynomial en la taille des données.

Le théorème suivant permet de situer ces différentes classes les unes par rapport aux autres.

Théorème n° B.4.2.I [Classes des problèmes de décision]

$$\mathcal{L} - space \subseteq \mathcal{P} \quad (\text{B.4.2.I})$$

$$\mathcal{NP} \subseteq \mathcal{P} - space \quad (\text{B.4.2.II})$$

$$(\text{B.4.2.III})$$

Démonstration n° B.4.2.I [Classes des problèmes de décision] Cette démonstration est disponible dans [44].

La figure B.4.2.I représente l'ensemble des relations (démonstrées ou supposées) entre les différentes classes des problèmes de décision.

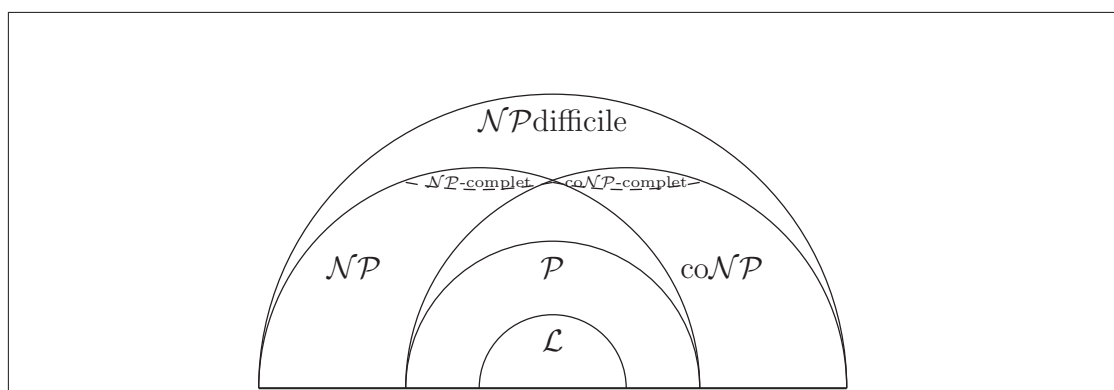


Fig. B.4.2.I: Diagramme de Venn des classes des problèmes de décision

Bien que cette classification est satisfaisante lorsqu'on considère des problèmes de décision, elle est insuffisante pour classifier les problèmes de recherche.

B.5 Problème de recherche

La hiérarchie polynomiale des classes de problèmes a pour but de hiérarchiser les classes de problèmes comprises entre \mathcal{NP} et $\mathcal{P} - space$. À cette attention, on considère l'hypothèse théorique et irréaliste selon laquelle on dispose d'un oracle puissant et bienveillant qui nous permet de résoudre un problème en temps constant. Cet artifice nous permet de définir de nouvelles classes de problèmes.

Définition n° B.5.0.II [Classes de problèmes]

\mathcal{P}^C est l'ensemble des problèmes pouvant être résolus de manière déterministe en temps polynomial en disposant d'un oracle pour un problème de classe C .

\mathcal{NP}^C est l'ensemble des problèmes pouvant être résolus de manière non-déterministe en temps polynomial en disposant d'un oracle pour un problème de classe C .

Le théorème suivant permet de situer ces différentes classes les unes par rapport aux autres.

Théorème n° B.5.0.II [Classes de problèmes]

$$\mathcal{NP} \subseteq \mathcal{P}^{\mathcal{NP}} \quad (\text{B.5.0.IV})$$

$$\text{co}\mathcal{NP} \subseteq \mathcal{P}^{\mathcal{NP}} \quad (\text{B.5.0.V})$$

$$(\mathcal{NP} \cup \text{co}\mathcal{NP}) \subseteq \mathcal{P}^{\mathcal{NP}} \quad (\text{B.5.0.VI})$$

Démonstration n° B.5.0.II [Classes de problèmes] *Trivial.*

Naturellement on peut considérer des oracles qui résolvent de manière déterministe en temps polynomial ces problèmes et récursivement.

Définition n° B.5.0.III [Hiérarchie polynomiale]

La **hiérarchie polynomiale** PH est constituée de la séquence des classes suivantes :

- $\Delta_0\mathcal{P} = \Sigma_0\mathcal{P} = \Pi_0 = \mathcal{P}$;
- $\forall i \geq 0$ on a :
 - $\Delta_{i+1}\mathcal{P} = \mathcal{P}^{\Sigma_i\mathcal{P}}$;
 - $\Sigma_{i+1}\mathcal{P} = \mathcal{NP}^{\Sigma_i\mathcal{P}}$;
 - $\Pi_{i+1}\mathcal{P} = \text{co}\mathcal{NP}^{\Sigma_i\mathcal{P}}$;
 - $PH = \bigcup_{i \geq 0} \Sigma_i\mathcal{P}$

La figure B.5.0.II représente l'ensemble des relations (démonstrées ou supposées) entre les différentes classes de problèmes.

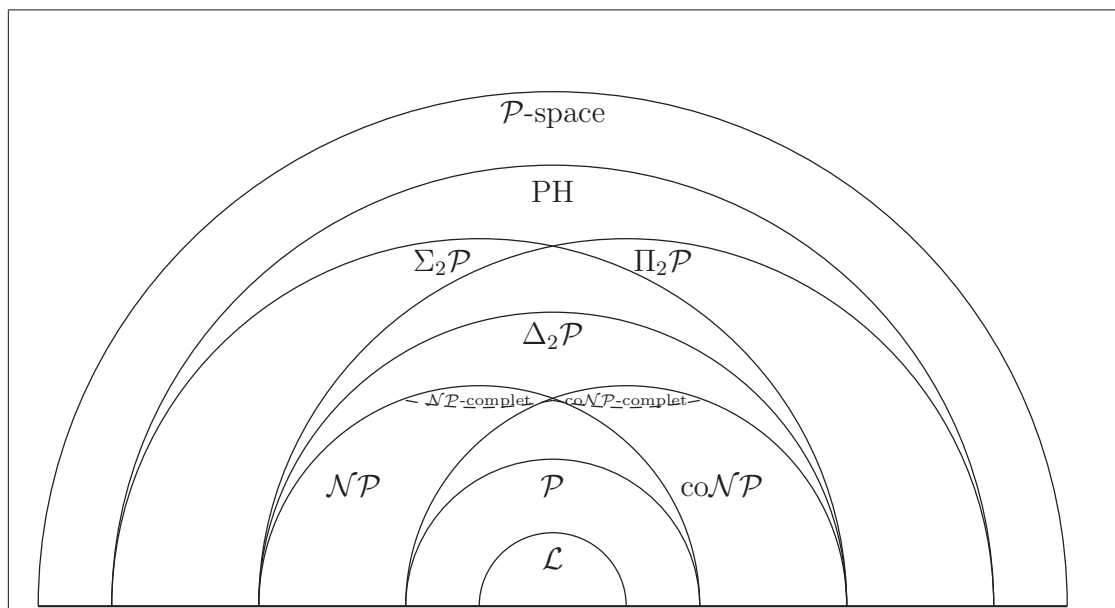


Fig. B.5.0.II: Diagramme de Venn de la hiérarchie polynomiale

B.6 Synthèse

Nous avons défini ici ce qu'est un algorithme et présenté les outils mathématiques utiles à leur analyse. Les notions de bornes permettent d'estimer l'ordre de grandeur de leur complexité en temps et en espace. L'analyse de l'efficacité des algorithmes permet de classer les problèmes de décision et de recherche selon leur complexité intrinsèque en temps et en espace.

« Les mathématiques sont la science de l'ordre et de la mesure. »

Descartes *Regulae*

Annexe C

Relation d'ordre

Sommaire

C.1 Introduction	249
C.2 Relations binaires	250
C.3 Relations d'ordre	250
C.4 Conclusion	252

C.1 Introduction

En mathématique, la notion de relation est une abstraction des liaisons qui existent dans le monde réel entre des choses, des grandeurs ou des phénomènes. Une relation sur un ensemble est une proposition qui lie certains de ces éléments. Par exemple, on peut définir dans un groupe de personnes une relation « Alice aime Bob ». Ainsi, une relation binaire est définie comme un ensemble de couples.¹ La notion d'ordre met en relation des objets d'un même ensemble afin de les comparer selon une certaine précedence. Elle précise si un objet est supérieur ou inférieur à un autre, voire s'il est impossible de les comparer.

Nous allons dans un premier temps définir les propriétés fondamentales qui permettent de caractériser une relation binaire (cf section C.2). Les différentes relations d'ordre sont définies et peuvent être comparées à partir de ces propriétés (cf section C.3).

¹Nous n'envisagerons dans ce chapitre que les relations binaires. Par exemple, « Léo est le fils d'Alice et de Bob » est une relation ternaire.

C.2 Relations binaires

Une relation binaire sur un ensemble E et un ensemble F est un sous-ensemble de $E \times F$, où $E \times F$ est le produit cartésien de E et F . Cette relation binaire est constituée d'un ensemble de couples de la forme (x, y) , noté xRy .

Une relation binaire peut être considérée comme une fonction de $E \times F$ à valeur dans l'ensemble des booléens. Les propriétés suivantes sont fondamentales :

Définition n° C.2.0.IV [Relation sur deux ensembles]

La relation binaire R sur E et F est dite :

- **surjective à gauche** ssi $\forall x \in E \exists y \in F xRy$;
- **surjective à droite** ssi $\forall y \in F \exists x \in E xRy$;
- **injective** ssi $\forall x, z \in E \forall y \in F (xRy \wedge zRy \Rightarrow x = z)$;
- **univoque** ssi $\forall x \in E \forall y, z \in F (xRy \wedge xRz \Rightarrow y = z)$.

Une fonction est une relation binaire univoque. Une application est une fonction qui est surjective à gauche.

Une relation binaire R sur un ensemble E est un sous-ensemble du produit cartésien $E \times E$. Elle peut être dotée des propriétés fondamentales suivantes :

Définition n° C.2.0.V [Relation sur un ensemble]

La relation binaire R sur E est dite :

- **réflexive** ssi $\forall x \in E xRx$;
- **antiréflexive** ssi $\forall x, y \in E (xRy \Rightarrow x \neq y)$;
- **symétrique** ssi $\forall x, y \in E (xRy \Rightarrow yRx)$;
- **asymétrique** si $\forall x, y \in E (xRy \Rightarrow \neg yRx)$;
- **antisymétrique** ssi $\forall x, y \in E (xRy \wedge yRx \Rightarrow x = y)$;
- **transitive** ssi $\forall x, y, z \in E (xRy \wedge yRz \Rightarrow xRz)$;
- **totale** ssi $\forall x, y \in E (xRy \vee yRx)$.

C'est à partir de ces propriétés que la notion d'ordre est formalisée.

C.3 Relations d'ordre

La notion d'ordre permet de comparer des objets selon une certaine précédence. Elle précise si un objet est supérieur ou inférieur à un autre, voire s'il est impossible de les comparer.

Une relation de pré-ordre est une relation binaire sur un ensemble défini de la manière suivante :

|| **Définition n° C.3.0.VI [Relation de pré-ordre]**

|| *R est une relation de **pré-ordre** ssi R est une relation réflexive et transitive.*

Une relation d'ordre est obtenue à partir d'une relation de pré-ordre à laquelle on adjoint l'antisymétrie.

|| **Définition n° C.3.0.VII [Relation d'ordre]**

|| *R est une relation d'**ordre** ssi R est une relation réflexive, transitive et antisymétrique.*

Une relation de pré-ordre (respectivement d'ordre) peut être totale ou partielle selon si tous les éléments sont comparables ou non (cf définition C.2.0.V).

Une relation d'ordre strict est obtenue à partir d'une relation d'ordre à laquelle on ôte la réflexivité :

|| **Définition n° C.3.0.VIII [Relation d'ordre strict]**

|| *R est une relation asymétrique et transitive.*

En d'autres termes, si \leq est une relation d'ordre, la relation d'ordre stricte associée $<$ est définie par : $x < y \Leftrightarrow x \leq y \wedge x \neq y$.

Le tableau C.3.0.I synthétise ces définitions. D'une part, on vérifie qu'une relation d'ordre est une relation de pré-ordre qui est antisymétrique. D'autre part, une relation d'ordre est obtenue à partir d'une relation d'ordre stricte en ajoutant la réflexivité.

L'exemple suivant permet d'illustrer la différence entre un ordre et un pré-ordre.

Exemple n° C.3.0.I [Relation d'ordre/pré-ordre] Soit $G = (S, A)$ un graphe orienté. Un chemin entre deux sommets $a, b \in S$ est une suite finie de n sommets (s_1, \dots, s_n) telle que $s_1 = a$, $s_2 = b$ et $\forall i \in [1, n - 1]$ $(s_i, s_{i+1}) \in A$. On dit que b est accessible depuis a . Cette relation d'accessibilité correspond à la fermeture réflexo-transitive du graphe. Cette relation est un pré-ordre. Si le graphe est sans circuit alors cette relation devient un ordre.

C.4 Conclusion

En mathématique, les différentes relations d'ordre sont des relations binaires sur un ensemble qui vérifient différentes propriétés. Elles permettent de classer des éléments, depuis les « meilleurs » jusqu'aux « moins bons », en tolérant (ou non) les *ex æquo*. Dans un pré-ordre (relation réflexive et transitive), des *ex æquo* sont possibles. Dans un ordre (relation réflexive, transitive et antisymétrique), il n'y a pas d'*ex æquo*. Un pré-ordre (respectivement un ordre) total est un pré-ordre (respectivement un ordre) dans lequel les éléments sont toujours comparables.

	réflexive	antiréflexive	symétrique	asymétrique	antisymétrique	transitive	totale
pré-ordre	X					X	
pré-ordre total	X					X	X
ordre	X				X	X	
ordre total	X				X	X	X
ordre strict				X		X	

Tab. C.3.0.I: Relations d'ordre

« Ce qui fait l'homme de science, ce n'est pas la possession de connaissances, d'irréfutables vérités, mais la quête obstinée et audacieusement critique de la vérité. »

Logique de la découverte scientifique de Karl Popper

Annexe D

Preuves

Sommaire

D.1	Introduction	255
D.2	Terminaison d'une demande d'information	255
D.3	Correction d'une demande d'information	258
D.4	Complétude d'une demande d'information	258
D.5	Terminaison des persuasions	259
D.6	Correction d'une persuasion	260

D.1 Introduction

Dans ce chapitre sont regroupées l'ensemble des démonstrations proposées dans le cadre du modèle DIAL.

D.2 Terminaison d'une demande d'information

Théorème n° D.2.0.III [Terminaison d'une demande d'information]
Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur le thème r_0 et qui emploie un protocole de demande d'information à réponses uniques. Le dialogue ($h \in Z$) est fini.

L'existence et l'unicité de la réponse à un coup, et ce, quelle que soit la règle de séquençage envisagée, ont été argumentées dans la section 9.5. Ses propriétés sont nécessaires pour la démonstration de ce théorème.

Démonstration n° D.2.0.III [Terminaison d'une demande d'information]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique multi-agents qui porte sur le thème r_0 et qui emploie un protocole de demande d'information à réponses uniques. Considérons l'historique $h \in H$.

Un dialogue débute par une question de l'initiateur ($\text{question}(r_0)$). Le partenaire peut toujours répondre à une question :

- soit par un aveu d'ignorance ($\text{unknow}(r_0)$) qui met fin à la ligne persuasive dans la situation de jeu 2.1. Comme le protocole est à réponses uniques, ce coup met fin au dialogue ($h \in Z$);
- soit par une affirmation ($\text{assert}(\neg r_0)$) dans la situation de jeu 2.2^{init};
- soit par une affirmation ($\text{assert}(r_0)$) dans la situation de jeu 2.3^{init}.

Les situations de jeu 2.2^{init} et 2.3^{init} sont équivalentes par symétrie sur le contenu de l'affirmation qui précède. Quitte à intervertir ces contenus, nous nous intéresserons uniquement à l'historique qui suit la situation de jeu 2.3^{init}.

L'initiateur peut toujours répondre à une affirmation ($\text{assert}(r_0)$) :

- soit par un accueil défavorable ($\text{refuse}(r_0)$) qui met fin à la ligne persuasive et donc au dialogue dans la situation de jeu 3.1 ($h \in Z$);
- soit par un accueil favorable ($\text{accept}(r_0)$) qui, pour la même raison, met fin au dialogue dans la situation de jeu 3.2 ($h \in Z$);
- soit par une mise en doute ($\text{challenge}(r_0)$) dans la situation de jeu 3.3^{part}.

Le partenaire peut toujours répondre à une mise en doute ($\text{challenge}(r_0)$) :

- soit par un abandon ($\text{withdraw}(r_0)$) qui met fin au dialogue dans la situation de jeu 4.1 ($h \in Z$);
- soit par l'affirmation d'un argument ($\text{assert}(H)$, avec $H \vdash_{\mathcal{U}} r_0$) dans la situation de jeu 4.2^{init}.

Les situations de jeu 2.3^{init} et 4.2^{init} sont équivalentes par récursion sur le contenu de l'affirmation qui précède.

La redondance d'information dans les affirmations d'une même ligne persuasive est interdite et par conséquent la présence de boucles dans un dialogue aussi. En outre, la théorie personnelle étendue $\mathcal{T}_{\text{part}}^*$ est finie parce que les théories personnelles des joueurs ($\mathcal{T}_{\text{part}}$ et $\mathcal{T}_{\text{init}}$) sont finies et donc le tableau d'engagement $CS_{\text{init}}^{\text{part}}$. Par conséquent, la récursion est finie et donc le dialogue se termine après un nombre fini de coups. \square

D.3 Correction d'une demande d'information

D'après le théorème suivant, une demande d'information est correcte si l'initiateur ne dispose initialement d'argument ni en faveur ni en défaveur du thème et si le partenaire est convaincu par le thème.

Théorème n° D.3.0.IV [Correction d'une demande d'information]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur le thème r_0 et qui emploie un protocole de demande d'information à réponses uniques. Si la situation initiale est telle que :

– l'initiateur ne dispose initialement d'aucun argument en faveur ou en défaveur de r_0 :

$$\forall A \in \mathcal{A}(\mathcal{T}_{\text{init}}) \text{ conclusion}(A) \neq r_0 \wedge \text{conclusion}(A) \neq \neg r_0$$

– le partenaire est initialement convaincu de r_0 :

$$\exists A \in \mathcal{S}_{\text{part}} \text{ conclusion}(A) = r_0$$

alors la demande d'information est correcte.

Afin de simplifier la lecture de ce théorème, nous supposons le partenaire convaincu de r_0 . Nous aurions pu tout aussi bien considérer le partenaire convaincu du contraire. Il suffit de remplacer dans les démonstrations suivantes r_0 par $\neg r_0$.

Démonstration n° D.3.0.IV [Correction d'une demande d'information]

Soit

$DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur le thème r_0 et qui emploie un protocole de demande d'information à réponses uniques.

Un dialogue débute par une question sur le thème ($\text{question}(r_0)$). Suite à cette question, le partenaire peut répondre d'après la règle de « Question/Réponse » (cf table 9.5.1) :

- soit par une confirmation ($\text{assert}(r_0)$);
- soit par une infirmation ($\text{assert}(\neg r_0)$);
- soit par un aveu d'ignorance ($\text{unknow}(r_0)$).

Le partenaire est convaincu de r_0 . Il dispose d'un argument en faveur de r_0 qu'il considère acceptable. Le partenaire est prévenant et coopératif. Par conséquence, il ne peut qu'affirmer r_0 . La situation de jeu 2.3^{init} est atteinte. Le tableau d'engagement correspondant est mis à jour par l'initiateur : $CS_{\text{part}}^{\text{init}}(h_2) = CS_{\text{part}}^{\text{init}}(h_1) \cup \{r_0\}$.

Suite à cette affirmation, l'initiateur peut répondre d'après la règle d'« Affirmation/Accueil » (cf table 9.5.2) :

- soit par un accueil favorable ($\text{accept}(r_0)$);
- soit par un accueil défavorable ($\text{refuse}(r_0)$);
- soit par une mise en doute ($\text{unknow}(r_0)$).

L'initiateur ne dispose initialement d'aucun argument en faveur ou en défaveur du thème. Il ne dispose donc, dans cette situation de jeu, que d'un argument trivial en faveur de cette thèse : $\exists A = (\{r_0\}, r_0) \in \mathcal{A}(CS_{\text{part}}^{\text{init}})$. Par conséquence, il ne peut que mettre en

doute r_0 et ce quelles que soient sa tactique argumentative d'acceptation et sa tactique de réponse à une affirmation. La situation de jeu 3.3^{part} est atteinte.

Suite à cette mise en doute, le partenaire peut répondre d'après la règle de « Mise en doute/Réponse » (cf table 9.5.3) :

- soit par une justification ($assert(H)$) tel que $H \vdash_{\cup} r_0$;
- soit par un abandon ($withdraw(r_0)$).

Si le partenaire abandonne alors le dialogue est clos et le témoin dispose d'un argument acceptable en faveur de r_0 ($provable^{h_4}(r_0), h_4 \in Z$). Si le partenaire se justifie alors il diffuse un argument non trivial en faveur de r_0 . Soit l'initiateur abdique à l'une des justifications du partenaire et le témoin dispose d'un argument acceptable en faveur de r_0 ($provable^h(r_0), h \in Z$). Soit l'initiateur attaque l'une des justifications du partenaire. Le témoin est alors convaincu par l'une des thèses. En d'autres termes, le protocole est correct. \square

D.4 Complétude d'une demande d'information

D'après le théorème suivant, une demande d'information est complète si l'initiateur est initialement ignorant et si le partenaire est convaincu par le thème.

Théorème n° D.4.0.V [Complétude d'une demande d'information]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, witness, H, T, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur le thème r_0 et qui emploie un protocole de demande d'information à réponses uniques. Si la situation initiale est telle que :

- l'initiateur est initialement ignorant :

$$\mathcal{T}_{init} = \mathcal{T}_{\Omega_A}$$

- le partenaire est initialement convaincu de r_0 :

$$\exists A \in \mathcal{S}_{part}^* conclusion(A) = r_0$$

alors la demande d'information est complète.

Pour les mêmes raisons évoquées précédemment, nous supposons que le partenaire est convaincu de r_0 .

Démonstration n° D.4.0.V [Complétude d'une demande d'information]

Soit

$DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, witness, H, T, convention, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur le thème r_0 et qui emploie un protocole de demande d'information à

réponses uniques.

On suppose que l'initiateur est ignorant dans la situation initiale. En d'autres termes, il ne dispose pas de connaissances propres. La théorie argumentative de l'agent omniscient et la théorie argumentative étendue du partenaire dans la situation initiale sont donc identiques. En conséquence, l'agent omniscient est convaincu de r_0 . Soit $h \in Z$ un dialogue. Étant donné la situation initiale et les tactiques de jeu des joueurs, la situation de jeu 3.3^{part} est atteinte. Si le partenaire abandonne, alors le dialogue est clos et le témoin dispose d'un argument acceptable en faveur de r_0 (provable^{h₄}(r_0), $h_4 \in Z$). Si le partenaire se justifie alors il diffuse un argument non trivial en faveur de r_0 . Étant donné que l'initiateur est initialement ignorant, soit l'initiateur abdique face à l'une des justifications du partenaire, soit le témoin abandonne. Au terme du dialogue, le témoin dispose d'un argument acceptable en faveur de r_0 (provable^h(r_0), $h \in Z$). En d'autres termes, la demande d'information est complète. \square

D.5 Terminaison des persuasions

Théorème n° D.5.0.VI [Terminaison des persuasions]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur le thème r_0 et qui emploie un protocole de persuasion à réponses uniques. Tous les dialogues ($h \in Z$) sont finis.

L'existence et l'unicité de la réponse à un coup, et ce quelle que soit la règle de séquençage envisagée, ont été argumentées dans la section 9.5. Ses propriétés sont nécessaires pour la démonstration suivante.

Démonstration n° D.5.0.VI [Terminaison des persuasions] Soit

$DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur le thème r_0 et qui emploie un protocole de persuasion à réponses uniques. Un dialogue débute par une question de l'initiateur ($\text{question}(r_0)$). Le partenaire peut toujours répondre à une question :

- soit par un aveu d'ignorance ($\text{unknow}(r_0)$) qui met fin à la ligne persuasive dans la situation de jeu 2.1 Comme le protocole est à réponses uniques, le coup met fin au dialogue ($h \in Z$);
- soit par une infirmation ($\text{assert}(\neg r_0)$) dans la situation de jeu 2.2^{init};
- soit par une confirmation ($\text{assert}(r_0)$) dans la situation de jeu 2.3^{init}.

Les situations de jeu 2.2^{init} et 2.3^{init} sont équivalentes par symétrie sur le contenu de l'affirmation qui précède. Quitte à intervertir ces contenus, nous nous intéresserons uniquement au déroulement de l'historique qui suit la situation de jeu 2.3^{init}.

L'initiateur peut toujours répondre à une affirmation ($\text{assert}(r_0)$) :

- soit par une réfutation ($\text{reject}(\neg r_0)$) dans la situation de jeu 3.1^{part} ;
- soit par un accueil favorable ($\text{accept}(r_0)$) qui met fin à la ligne persuasive et donc au dialogue dans la situation de jeu 3.2 ;
- soit par une mise en doute ($\text{challenge}(r_0)$) dans la situation de jeu 3.3^{part}.

Envisageons les deux situations de jeux suivantes : 3.3^{part} et 3.1^{part}.

Dans la situation de jeu 3.3^{part}, le partenaire peut toujours répondre à une mise en doute ($\text{challenge}(r_0)$) :

- soit par un abandon ($\text{withdraw}(r_0)$) qui met fin au dialogue ($h \in Z$) ;
- soit par une justification ($\text{assert}(H)$, avec $H \vdash_{\mathcal{U}} h$) dans la situation de jeu 4.4^{init}.

Les situations de jeu 2.3^{init} et 4.4^{init} sont équivalentes par récursion sur le contenu propositionnel de l'affirmation qui précède.

Dans la situation de jeu 3.1^{part}, l'initiateur peut toujours répondre à cette affirmation ($\text{reject}(r_0) \Leftrightarrow \text{assert}(\neg r_0)$) :

- soit par un accueil défavorable ($\text{assert}(r_0)$) interdit par la règle de non-redondance ;
- soit par un accueil favorable ($\text{accept}(\neg r_0)$) qui met fin au dialogue ($h \in Z$) ;
- soit par une mise en doute ($\text{challenge}(\neg r_0)$) dans la situation de jeu 4.1^{init}.

L'initiateur peut toujours répondre à une mise en doute ($\text{challenge}(\neg r_0)$) :

- soit par un abandon ($\text{withdraw}(\neg r_0)$) qui met fin au dialogue ($h \in Z$) ;
- soit par une justification ($\text{assert}(H)$, avec $H \vdash_{\mathcal{U}} \neg r_0$) dans la situation de jeu 5.1^{part}.

Les situations de jeu 3.1^{part} et 5.1^{part} sont équivalentes par récursion sur le contenu de l'affirmation précédente.

En résumé, les situations de jeu 5.1^{part} et 4.4^{init} sont les seules sur des historiques en cours. Elles sont équivalentes, par récursion sur le contenu des affirmations précédentes, à des situations de jeu qui les précèdent dans leur ligne persuasive. De plus, la redondance d'information dans les affirmations dans une même ligne persuasive sont interdites et par conséquent la présence de boucles dans un dialogue. En outre, $\mathcal{T}_{\text{part}}$ et $\mathcal{T}_{\text{init}}$ sont finies et donc les tableaux d'engagement $CS_{\text{init}}^{\text{part}}$ et $CS_{\text{part}}^{\text{init}}$ sont également finis. Par conséquent, suite aux situations de jeu 3.1^{part} et 3.3^{part}, la récursion est finie et donc le dialogue se termine après un nombre fini de coups. \square

D.6 Correction d'une persuasion

D'après le théorème suivant, une persuasion est correcte si l'initiateur et le partenaire ont des avis différents sur le thème.

Théorème n° D.6.0.VII [Correction d'une persuasion]

Soit $DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur le thème r_0 et qui emploie un protocole de persuasion à réponses uniques.

Si la situation initiale des participants est telle que :

- l'initiateur est convaincu de $\neg r_0$:

$$\exists A \in \mathcal{S}_{\text{init}}^* \text{conclusion}(A) = r_0$$

- le partenaire est convaincu de r_0 :

$$A \in \mathcal{S}_{\text{part}}^* \text{conclusion}(A) = r_0$$

alors la persuasion est correcte.

Démonstration n° D.6.0.VII [Correction d'une persuasion] Soit

$DS_{\Omega_M}(r_0, \mathcal{AT}_{\Omega_A}) = \langle N, \text{witness}, H, T, \text{convention}, Z, (u_p)_{p \in N} \rangle$ un système dialectique qui porte sur le thème r_0 et qui emploie un protocole de persuasion à réponses uniques. Un dialogue débute par une question sur le thème du dialogue ($\text{question}(r_0)$). Suite à cette question, le partenaire peut répondre d'après la règle de « Question/Réponse » (cf table 9.5.1) :

- soit par une confirmation ($\text{assert}(r_0)$);
- soit par une infirmation ($\text{assert}(\neg r_0)$);
- soit par un aveu d'ignorance ($\text{unknow}(r_0)$).

Le partenaire convaincu de r_0 : il dispose d'un argument acceptable en faveur de r_0 . Le partenaire est prévenant et coopératif. Par conséquence, il ne peut qu'affirmer r_0 . La situation de jeu 2.3^{init} est atteinte. Le tableau d'engagement correspondant est mis à jour par l'initiateur : $CS_{\text{part}}^{\text{init}}(h_2) = CS_{\text{part}}^{\text{init}}(h_1) \cup \{r_0\}$.

Suite à cette affirmation, l'initiateur peut répondre d'après la règle d'« Affirmation/Réfutation » (cf table 9.5.2) :

- soit par un accueil favorable ($\text{accept}(r_0)$);
- soit par une réfutation ($\text{reject}(r_0)$);
- soit par une mise en doute ($\text{unknow}(r_0)$).

Or l'initiateur est, au début de la persuasion, convaincu de $\neg r_0$. Il dispose d'un argument acceptable en faveur de r_0 . Il dispose également d'un nouvel argument trivial en faveur de $\neg r_0$. L'initiateur est prévenant et ouvert d'esprit. Par conséquence, il ne peut que réfuter r_0 . La situation de jeu 3.1^{part} est atteinte. Le tableau d'engagement correspondant est mis à jour par l'initiateur : $CS_{\text{init}}^{\text{part}}(h_3) = CS_{\text{part}}^{\text{init}}(h_2) \cup \{\neg r_0\}$.

Suite à cette affirmation, le partenaire peut répondre d'après la règle d'« Affirmation/Réfutation » (cf table 9.5.2) :

- soit par une réfutation ($\text{assert}(r_0)$) interdite par la règle de non-redondance ;

- soit par un accueil favorable ($\text{accept}(\neg r_0)$);
- soit par une mise en doute ($\text{challenge}(\neg r_0)$).

Or le partenaire, convaincu de r_0 au début de la persuasion, dispose dans cette situation de jeu d'un argument acceptable en faveur de r_0 . Il dispose également d'un argument trivial en faveur de $\neg r_0$. Par conséquent, il ne peut que mettre en doute r_0 . La situation de jeu 4.1^{init} est atteinte.

- Si la règle de non-redondance contraint l'initiateur à abandonner alors la situation de jeu 5.2 est atteinte. L'agent témoin possède un argument trivial en faveur de r_0 et un argument trivial pour $\neg r_0$. Ces arguments s'attaquent mutuellement. Comme la source de ces arguments diffère, seul l'un d'eux est acceptable et donc l'agent témoin est enclin à l'une des thèses :
 - si $\text{init} \prec_{\text{witness}} \text{part}$ et donc le témoin est convaincu de r_0 ;
 - sinon $\text{part} \prec_{\text{witness}} \text{init}$ alors le témoin est convaincu de $\neg r_0$.
- Si l'initiateur possède un argument non-trivial pour $\neg r_0$, ses stratégies argumentatives et conventionnelles peuvent l'autoriser à se justifier. La situation de jeu 5.1^{part} est alors atteinte.

Dans les situations de fin de jeu 6.2 et 7.1, l'argument trivial en faveur de $\neg r_0$ est le seul argument acceptable par le témoin. Ce dernier est alors convaincu de $\neg r_0$. Les situations de jeu restantes sont équivalentes par récursion sur le contenu propositionnel du message précédent ;

Par conséquent, le témoin est convaincu de l'une des thèses des joueurs quelle que soit la manière dont le dialogue est clos. En d'autres termes, la persuasion est correcte. \square

« Un bon exemple vaut mieux
qu'un long discours. »

Adage

Annexe E

Exemples

Sommaire

E.1	Introduction	263
E.2	Demande d'information correcte et complète	263
E.3	Demande d'information correcte et incomplète	269
E.4	Persuasion	271

E.1 Introduction

Deux adhérents d'une association loi 1901 doivent, au cours d'une assemblée générale, se prononcer pour désigner le futur président de cette association. Un tel débat peut être modélisé à l'aide d'un système dialectique dont le thème est « *president(jose)* ».

Nous proposons ici deux exemples de demande d'information. La première demande d'information est correcte et complète (cf section E.2). La seconde demande d'information est correcte mais incomplète (cf section E.3). Nous proposons également un exemple de persuasion (cf section E.4). Cette persuasion est correcte.

E.2 Demande d'information correcte et complète

Considérons une première demande d'information. L'initiateur est ignorant au début du dialogue. Le partenaire est un partisan du candidat jose. La demande d'information présentée ici consiste en une séquence de six coups dialogiques.

D'une part, on vérifie que la demande d'information est correcte. L'initiateur ainsi que le témoin sont, au terme du dialogue, des partisans du candidat jose comme le

partenaire. D'autre part, la demande d'information est complète. En effet, le témoin est au terme du dialogue partisan du candidat jose comme l'agent omniscient.

Situation initiale

Dans la situation de jeu 0^{init} , l'initiateur est ignorant. Les seules règles dont ils disposent sont des règles communes (cf figure E.2.0.I). Il ne dispose d'aucun argument en faveur ou en défaveur de l'un des candidats.

\ll^*_{init}	V^*_{init}	$\mathcal{T}^*_{\text{init}}$	\mathcal{T}_{Ω_A}
↑	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	
	v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$	
	$v^{\text{init}}_{\text{part}}$	\emptyset	

Fig. E.2.0.I: Théorie argumentative étendue de l'initiateur dans la situation initiale pour la première demande d'information

Le partenaire, représenté dans la figure E.2.0.II, dispose quant à lui d'un argument en faveur du candidat jose (B) et d'un argument en faveur du candidat jack (A_2). B est, contrairement à A_2 , un argument acceptable par le partenaire.

\ll^*_{part}	V^*_{part}	$\mathcal{T}^*_{\text{part}}$	\mathcal{T}_{Ω_A}
↑	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	
	v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$	
	v_3	$r_3(x) : \text{president}(x) \leftarrow \text{president_actuel}(x)$	\widehat{B} \swarrow A_2
	v_4	$r_4 : \text{escroc}(\text{jack}) \leftarrow$	
	v_5	$r_5 : \text{prison}(\text{jose}) \leftarrow$	
	$v^{\text{part}}_{\text{init}}$	$\text{CS}^{\text{part}}_{\text{init}} = \emptyset$	

Fig. E.2.0.II: Théorie argumentative étendue du partenaire dans la situation initiale pour la première demande d'information

Chacun des deux joueurs se considère plus compétent que son interlocuteur. En effet, la valeur $v^{\text{init}}_{\text{part}}$ (respectivement $v^{\text{part}}_{\text{init}}$) associée au tableau d'engagement $\text{CS}^{\text{init}}_{\text{part}}$

E.2. DEMANDE D'INFORMATION CORRECTE ET COMPLÈTE 265

(respectivement $CS_{\text{init}}^{\text{part}}$) est, pour l'initiateur (respectivement le partenaire), la valeur la moins prioritaire. Nous supposons que le partenaire coopératif est prévenant.

L'agent témoin ne dispose initialement d'aucune connaissance propre. Il est représenté dans la figure E.2. Il considère l'initiateur comme plus compétent que le partenaire. En effet, la valeur $v_{\text{part}}^{\text{witness}}$ associée au tableau d'engagement $CS_{\text{part}}^{\text{witness}}$ est moins prioritaire que la valeur $v_{\text{init}}^{\text{witness}}$ associée au tableau d'engagement $CS_{\text{init}}^{\text{witness}}$.

\ll_{witness}^*	V_{witness}^*	$\mathcal{T}_{\text{witness}}^*$	\mathcal{T}_{Ω_A}
↑	v_1		$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$
	v_2		$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$
	$v_{\text{init}}^{\text{witness}}$	\emptyset	
	$v_{\text{part}}^{\text{witness}}$	\emptyset	

Fig. E.2.0.III: Théorie argumentative étendue du témoin dans la situation initiale pour la première demande d'information

L'agent omniscient, représenté dans la figure E.2, dispose quant à lui de la connaissance de l'initiateur et de celle du partenaire. Il considère l'initiateur plus compétent que le partenaire. En effet, les valeurs propres à l'initiateur ($v \in V_{\text{init}}^{\text{omni}}$) sont plus prioritaires que les valeurs propres au partenaire ($V_{\text{part}}^{\text{omni}}$). B est, contrairement à A_2 , un argument acceptable. L'agent omniscient est un partisan du candidat jose.

\ll_{omni}	V_{omni}	$\mathcal{T}_{\text{omni}}$	\mathcal{T}_{Ω_A}
↑	v_1		$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$
	v_2		$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$
	$v_6 \in V_{\text{init}}^{\text{omni}}$	\emptyset	
	$v_3 \in V_{\text{part}}^{\text{omni}}$	$r_3(x) : \text{president}(x) \leftarrow \text{president_actuel}(x)$	
	$v_4 \in V_{\text{part}}^{\text{omni}}$	$r_4 : \text{escroc}(\text{jack}) \leftarrow$	
	$v_5 \in V_{\text{part}}^{\text{omni}}$	$r_5 : \text{prison}(\text{jose}) \leftarrow$	

$(\overline{B}) \rightarrow A_2$

Fig. E.2.0.IV: Théorie argumentative de l'agent omniscient

Une demande d'information débute par une question.

Réponse à une question

Le coup d'initialisation suivant est émis par l'initiateur à l'intention du partenaire :

$$\text{move}_1 = \langle \text{init}, \text{part}, \emptyset, \text{ISP}, \text{question}(\text{president}(\text{jose}) \leftarrow) \rangle$$

D'après la règle de « Question/Réponse », le partenaire peut répondre : soit par une confirmation ($\text{assert}(\text{president}(\text{jose}) \leftarrow)$), soit par une infirmation ($\text{assert}(\neg \text{president}(\text{jose}) \leftarrow)$), soit par un aveu d'ignorance ($\text{unknow}(\text{president}(\text{jose}) \leftarrow)$).

Dans la situation de jeu 1^{part} , le partenaire dispose d'arguments en faveur de chacun des candidats. En conséquence, les conditions d'énonciation de la confirmation et de l'infirmation sont vérifiées. De plus, le partenaire est prévenant. En conséquence, la tactique d'énonciation de la confirmation est vérifiée contrairement à la tactique d'énonciation de l'infirmation. D'après l'algorithme 9.5.1, le partenaire qui est coopératif répond par une confirmation.

Réponse à une première affirmation

Le coup de réponse suivant est émis par le partenaire à l'intention de l'initiateur :

$$\text{move}_2 = \langle \text{part}, \text{init}, \text{move}_1, \text{ISP}, \text{assert}(\text{president}(\text{jose}) \leftarrow) \rangle$$

Les auditeurs, qu'il s'agisse de l'initiateur ou de l'agent témoin, mettent à jour le tableau d'engagement correspondant :

$$\begin{aligned} \text{CS}_{\text{part}}^{\text{witness}}(\text{move}_2) &= \text{CS}_{\text{part}}^{\text{witness}}(\text{move}_1) \cup \{\text{president}(\text{jose}) \leftarrow\} \\ \text{CS}_{\text{part}}^{\text{init}}(\text{move}_2) &= \text{CS}_{\text{part}}^{\text{init}}(\text{move}_1) \cup \{\text{president}(\text{jose}) \leftarrow\} \end{aligned}$$

D'après la règle d'« Affirmation/Accueil », l'initiateur peut répondre : soit par un accueil favorable ($\text{concede}(\text{president}(\text{jose}) \leftarrow)$), soit par un accueil défavorable ($\text{refuse}(\text{president}(\text{jose}) \leftarrow)$), soit par une mise en doute ($\text{challenge}(\text{president}(\text{jose}) \leftarrow)$).

Dans la situation de jeu 2^{init} , B' est le seul argument relatif à cette hypothèse dont dispose l'initiateur :

$$B' = (\{\text{president}(\text{jose}) \leftarrow\}, \text{president}(\text{jose}) \leftarrow)$$

L'argument B' est non seulement trivial mais sa prémisse est incluse dans le tableau d'engagement. En conséquence, cette hypothèse ne peut pas être concédée. Cette hypothèse est mise en doute.

Réponse à une première mise en doute

Le coup de réponse suivant est émis par l'initiateur à l'intention du partenaire :

$$\text{move}_3 = \langle \text{init}, \text{part}, \text{move}_2, \text{ISP}, \text{challenge}(\text{president}(\text{jose}) \leftarrow) \rangle$$

D'après la règle de « Mise en doute/Réponse », le partenaire peut répondre par une justification ($\text{assert}(H)$, avec $H \vdash_{\mathcal{U}} \text{president}(\text{jose}) \leftarrow$), ou par un abandon $\text{withdraw}(\text{president}(\text{jose}) \leftarrow)$.

Dans la situation de jeu 3^{part} , le partenaire dispose de l'argument suivant en faveur du candidat jose :

$$B = (\{r_4, r_{12}(\text{jack}), r_{22}(\text{jack}), r_{21}\}, \text{president}(\text{jose}) \leftarrow)$$

Chacun des éléments de la prémisse est soutenu par un argument trivial acceptable. En conséquence la condition d'énonciation ainsi que la tactique d'énonciation de la justification sont vérifiées. Cette hypothèse est justifiée.

Réponse à une seconde affirmation

Le coup de réponse suivant est émis par le partenaire à l'intention de l'initiateur :

$$\text{move}_4 = \langle \text{part}, \text{init}, \text{move}_3, \text{ISP}, \text{assert}(\{r_4, r_{12}(\text{jack}), r_{22}(\text{jack}), r_{21}\}) \rangle$$

Les auditeurs, qu'il s'agisse de l'initiateur ou de l'agent témoin, mettent à jour le tableau d'engagement correspondant :

$$\begin{aligned} \text{CS}_{\text{part}}^{\text{witness}}(\text{move}_4) &= \text{CS}_{\text{part}}^{\text{witness}}(\text{move}_3) \cup \{r_4, r_{12}(\text{jack}), r_{22}(\text{jack}), r_{21}\} \\ \text{CS}_{\text{part}}^{\text{init}}(\text{move}_4) &= \text{CS}_{\text{part}}^{\text{init}}(\text{move}_3) \cup \{r_4, r_{12}(\text{jack}), r_{22}(\text{jack}), r_{21}\} \end{aligned}$$

D'après la règle d' « Affirmation/Accueil », l'initiateur peut répondre : soit par un accueil favorable ($\text{concede}(\text{premise}(B))$), soit par un accueil défavorable ($\text{refuse}(\text{premise}(B))$), soit par une mise en doute ($\text{challenge}(h)$ avec $h \in \text{premise}(B)$).

Dans la situation de jeu 4^{init} , toutes les hypothèses à l'exception de r_4 sont soutenues par un argument trivial dont la prémisse est un élément de la théorie commune. La condition d'énonciation de cet ensemble d'hypothèses n'est pas vérifiée. En conséquence, l'hypothèse r_4 est mise en doute.

Réponse à une seconde mise en doute

Le coup de réponse suivant est émis par l'initiateur à l'intention du partenaire :

$$\text{move}_5 = \langle \text{init, part, move}_4, \text{ISP, challenge}(\text{escroc}(\text{jack}) \leftarrow) \rangle$$

D'après la règle de « Mise en doute/Réponse », le partenaire peut répondre par une justification ($\text{assert}(H)$, avec $H \vdash_{\cup} \text{escroc}(\text{jack}) \leftarrow$), ou par un abandon $\text{withdraw}(\text{escroc}(\text{jack}) \leftarrow)$.

Dans la situation de jeu 5^{part} , le partenaire dispose seulement d'un argument trivial en faveur de cette hypothèse. De plus, la règle de non-redondance lui interdit d'affirmer une nouvelle fois cette hypothèse. En conséquence, le partenaire abandonne.

Clôture du dialogue

Le coup de réponse suivant est émis par le partenaire à l'intention de l'initiateur :

$$\text{move}_6 = \langle \text{part, init, move}_5, \text{ISP, withdraw}(\text{escroc}(\text{jack}) \leftarrow) \rangle$$

Ce coup met un terme au dialogue.

Situation finale

Comme l'illustre la figure E.2, le témoin dispose finalement de deux arguments acceptables B et B' . Le témoin est au terme de ce dialogue un partisan du candidat jose.

$\llcorner_{\text{witness}}^*$	V_{witness}^*	$\mathcal{T}_{\text{witness}}^*$	\mathcal{T}_{Ω_A}
↑	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	
	v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$	
	$v_{\text{init}}^{\text{witness}}$	\emptyset	
	$v_{\text{part}}^{\text{witness}}$	$\text{president}(\text{jose}) \leftarrow$ $\text{escroc}(\text{jack}) \leftarrow$	$\overset{B'}{\curvearrowright} \underset{B}{\curvearrowleft}$

Fig. E.2.0.V: Théorie argumentative étendue du témoin au terme de la première demande d'information

E.3. DEMANDE D'INFORMATION CORRECTE ET INCOMPLÈTE

D'une part cette demande d'information est correcte. En effet, le témoin ainsi que l'initiateur sont, au terme de la demande d'information, partisans du candidat jose comme le partenaire. D'autre part, cette demande d'information est complète. En effet, le témoin est au terme du dialogue partisan du candidat jose comme l'agent omniscient. La seconde demande d'information proposée ci-dessous n'est pas complète.

E.3 Demande d'information correcte et incomplète

Considérons une seconde demande d'information. Le partenaire est un partisan du candidat jose. Contrairement à l'exemple précédent, l'initiateur n'est pas totalement ignorant. Malgré cela, le dialogue reste identique au précédent.

D'une part, on vérifie que le dialogue est correct. En effet, l'initiateur ainsi que le témoin sont, au terme du dialogue, des partisans du candidat jose comme le partenaire. D'autre part, la situation initiale des joueurs est telle que la demande d'information est incomplète. En effet, le témoin est au terme du dialogue partisan du candidat jose. Ce n'est pas le cas de l'agent omniscient.

Situation initiale

Dans la situation de jeu 0^{init} , l'initiateur n'est pas ignorant. Il dispose non seulement des règles communes mais également de la règle r_6 (cf figure E.3). Toutefois, il ne dispose d'aucun argument en faveur ou en défaveur d'un des candidats.

$\llcorner_{\text{init}}^*$	V_{init}^*	$\mathcal{T}_{\text{init}}^*$	\mathcal{T}_{Ω_A}
↑	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	
	v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$	
	v_6	$r_6(x) : \text{president_actuel}(\text{jack}) \leftarrow$	
	$v_{\text{part}}^{\text{init}}$	\emptyset	

Fig. E.3.0.VI: Théorie argumentative de l'initiateur dans la situation initiale pour la seconde demande d'information

Le partenaire est identique à l'exemple précédent. Il dispose initialement de la même connaissance. Cet agent est représenté dans la figure E.2.0.II. Le partenaire dispose

d'un argument en faveur du candidat jose (B) et d'un argument en faveur du candidat jack (A_2). L'ensemble $\{B\}$ est, contrairement à $\{A_2\}$ acceptable. Nous supposons ici que le partenaire est coopératif et prévenant. Comme précédemment, chacun des deux joueurs se considère plus compétent que son interlocuteur.

L'agent témoin est identique à l'exemple précédent. Il est initialement ignorant (cf figure E.2).

L'agent omniscient, représenté dans la figure E.3, dispose quant à lui de la connaissance de l'initiateur et de celle du partenaire. Comme dans l'exemple précédent, il considère l'initiateur comme plus compétent que le partenaire. Il dispose d'un argument supplémentaire A_1 . Ni l'initiateur ni le partenaire ne disposent de cet argument. Ce dernier est le résultat de l'union des théories personnelles. L'ensemble des arguments $\{A_1, A_2\}$ est acceptable contrairement à $\{B\}$. L'agent omniscient est partisan du candidat jack.

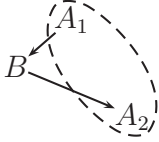
\llcorner_{omni}	V_{omni}	$\mathcal{T}_{\text{omni}}$	\mathcal{T}_{Ω_A}
↑	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	
	v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$	
	$v_6 \in V_{\text{init}}^{\text{omni}}$	$r_6 : \text{president_actuel}(\text{jack}) \leftarrow$	
	$v_3 \in V_{\text{part}}^{\text{omni}}$	$r_3(x) : \text{president}(x) \leftarrow \text{president_actuel}(x)$	
	$v_4 \in V_{\text{part}}^{\text{omni}}$	$r_4 : \text{escroc}(\text{jack}) \leftarrow$	
	$v_5 \in V_{\text{part}}^{\text{omni}}$	$r_5 : \text{prison}(\text{jose}) \leftarrow$	

Fig. E.3.0.VII: Théorie argumentative de l'agent omniscient de la seconde demande d'information

La demande d'information se déroule exactement de la même manière que précédemment.

Situation finale

Comme l'illustre la figure E.2, le témoin dispose finalement de deux arguments B et B' . $\{B, B'\}$ est un ensemble d'arguments acceptables. Le témoin est au terme de ce dialogue un partisan du candidat jose.

D'une part, on vérifie que le dialogue est correct. En effet, l'initiateur ainsi que le témoin sont, au terme du dialogue, des partisans du candidat jose comme le partenaire. D'autre part, la demande d'information est incomplète. En effet, le témoin

est au terme du dialogue partisan du candidat jose. Ce n'est pas le cas de l'agent omniscient.

E.4 Persuasion

Considérons une persuasion au travers de laquelle s'affrontent deux joueurs. L'initiateur est un partisan du candidat jack. Son partenaire est partisan du candidat jose. Le dialogue présenté ici consiste en une séquence de 7 coups.

D'une part, on vérifie que la persuasion est correcte. Le témoin est, au terme du dialogue, partisan de l'un des candidats. Son avis dépend du joueur qu'il considère le plus compétent.

Situation initiale

Dans la situation de jeu 0^{init} , l'initiateur est un partisan du candidat jack. A_2 est le seul argument relatif au thème dont dispose l'initiateur. Cet argument en faveur du candidat jack est selon lui acceptable (cf figure E.4).

$\llcorner_{\text{init}}^*$	V_{init}^*	$\mathcal{T}_{\text{init}}^*$	\mathcal{T}_{Ω_A}
↑	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	
	v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$	
	v_5	$r_5 : \text{prison}(\text{jose}) \leftarrow$	(\bar{A}_2)
	$v_{\text{part}}^{\text{init}}$	\emptyset	

Fig. E.4.0.VIII: Théorie argumentative étendue de l'initiateur dans la situation initiale pour la persuasion

Le partenaire est quant à lui représenté dans la figure E.4.0.IX. B est le seul argument relatif au thème du partenaire. Cet argument en faveur du candidat jose est selon lui acceptable.

Chacun des deux participants se considère plus compétent que son interlocuteur. En effet, la valeur $v_{\text{part}}^{\text{init}}$ (respectivement $v_{\text{init}}^{\text{part}}$) associée au tableau d'engagement $\text{CS}_{\text{part}}^{\text{init}}$ (respectivement $\text{CS}_{\text{init}}^{\text{part}}$) est pour l'initiateur (respectivement le partenaire) la moins prioritaire. Nous supposons ici que l'initiateur est ouvert d'esprit et que le partenaire est coopératif.

$\llcorner_{\text{part}}^*$	V_{part}^*	$\mathcal{T}_{\text{part}}^*$	\mathcal{T}_{Ω_A}
	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	
	v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$	
	v_3	$r_3(x) : \text{president}(x) \leftarrow \text{president_actuel}(x)$	
	v_4	$r_4 : \text{escroc}(\text{jack}) \leftarrow$	(\hat{B})
	$v_{\text{init}}^{\text{part}}$	\emptyset	

Fig. E.4.0.IX: Théorie argumentative étendue du partenaire dans la situation initiale pour la persuasion

Dans la situation initiale, l'agent témoin ne dispose pas d'autres connaissances que la théorie commune. On distingue deux cas de figures :

- cas 1 : il considère l'initiateur plus compétent que le partenaire. La valeur $v_{\text{part}}^{\text{witness}}$ associée au tableau d'engagement $\text{CS}_{\text{part}}^{\text{witness}}$ est alors moins prioritaire que la valeur $v_{\text{init}}^{\text{witness}}$ associée au tableau d'engagement $\text{CS}_{\text{init}}^{\text{witness}}$;
- cas 2 : il considère l'initiateur moins compétent que le partenaire. La valeur $v_{\text{part}}^{\text{witness}}$ associée au tableau d'engagement $\text{CS}_{\text{part}}^{\text{witness}}$ est alors plus prioritaire que la valeur $v_{\text{init}}^{\text{witness}}$ associée au tableau d'engagement $\text{CS}_{\text{init}}^{\text{witness}}$.

Une persuasion débute par une question sur le thème.

Réponse à une question

Le coup d'initialisation suivant est émis par l'initiateur à l'intention du partenaire :

$$\text{move}_1 = \langle \text{init}, \text{part}, \emptyset, \text{PP}, \text{question}(\text{president}(\text{jose}) \leftarrow) \rangle$$

D'après la règle de « Question/Réponse », le partenaire peut répondre : soit par une confirmation ($\text{assert}(\text{president}(\text{jose}) \leftarrow)$), soit par une infirmation ($\text{assert}(\neg\text{president}(\text{jose}) \leftarrow)$), soit par un aveu d'ignorance ($\text{unknow}(\text{president}(\text{jose}) \leftarrow)$).

Dans la situation de jeu 1^{part} , le partenaire dispose d'un seul argument en faveur du candidat jose. En conséquence, la tactique d'énonciation de la confirmation est vérifiée, que le partenaire soit prévenant ou confiant. De plus, la condition d'énonciation de l'infirmité n'est pas vérifiée. D'après l'algorithme 9.5.1, le partenaire qui est coopératif répond par une confirmation.

Réponse à une première affirmation

Le coup de réponse suivant est émis par le partenaire à l'intention de l'initiateur :

$$\text{move}_2 = \langle \text{part, init, move}_1, \text{PP, assert}(\text{president}(\text{jose}) \leftarrow) \rangle$$

Les auditeurs, qu'il s'agisse de l'initiateur ou de l'agent témoin, mettent à jour le tableau d'engagement correspondant :

$$\begin{aligned} \text{CS}_{\text{part}}^{\text{witness}}(\text{move}_2) &= \text{CS}_{\text{part}}^{\text{witness}}(\text{move}_1) \cup \{\text{president}(\text{jose}) \leftarrow\} \\ \text{CS}_{\text{part}}^{\text{init}}(\text{move}_2) &= \text{CS}_{\text{part}}^{\text{init}}(\text{move}_1) \cup \{\text{president}(\text{jose}) \leftarrow\} \end{aligned}$$

D'après la règle d' « Affirmation/Réfutation », l'initiateur peut répondre : soit par un accueil favorable ($\text{concede}(\text{president}(\text{jose}) \leftarrow)$), soit par une réfutation ($\text{reject}(\text{president}(\text{jose}) \leftarrow)$), soit par une mise en doute ($\text{challenge}(\text{president}(\text{jose}) \leftarrow)$).

Dans la situation de jeu 2^{init} , l'initiateur dispose de l'argument suivant en faveur de cette hypothèse :

$$B' = (\{\text{president}(\text{jose})\}, \text{president}(\text{jose}) \leftarrow)$$

L'argument B' est non seulement trivial mais sa prémisse est incluse dans le tableau d'engagement. En conséquence, la condition d'énonciation de l'accueil favorable n'est pas vérifiée. L'initiateur dispose également de l'argument A'_2 qui est un sous-argument de A_2 :

$$A'_2 = (\{r_5, r_{22}(\text{jose})\}, \neg\text{president}(\text{jose}) \leftarrow)$$

Selon l'initiateur, cet argument est acceptable. En conséquence, la tactique d'énonciation de la réfutation est vérifiée, que l'initiateur soit prévenant ou confiant. D'après l'algorithme 9.5.2, l'initiateur qui est ouvert d'esprit répond par une réfutation.

Réponse à une seconde affirmation

Le coup de réponse suivant est émis par l'initiateur à l'intention du partenaire :

$$\text{move}_3 = \langle \text{init, part, move}_2, \text{PP, assert}(\neg\text{president}(\text{jose}) \leftarrow) \rangle$$

Les auditeurs, qu'il s'agisse du partenaire ou de l'agent témoin, mettent à jour le tableau d'engagement correspondant :

$$\begin{aligned} \text{CS}_{\text{init}}^{\text{witness}}(\text{move}_3) &= \text{CS}_{\text{init}}^{\text{witness}}(\text{move}_2) \cup \{\neg\text{president}(\text{jose}) \leftarrow\} \\ \text{CS}_{\text{init}}^{\text{part}}(\text{move}_3) &= \text{CS}_{\text{init}}^{\text{part}}(\text{move}_2) \cup \{\neg\text{president}(\text{jose}) \leftarrow\} \end{aligned}$$

D'après la règle d' « Affirmation/Réfutation », le partenaire peut répondre : soit par un accueil favorable ($\text{concede}(\neg\text{president}(\text{jose}) \leftarrow)$), soit par une réfutation ($\text{reject}(\neg\text{president}(\text{jose}) \leftarrow)$), soit par une mise en doute ($\text{challenge}(\text{president}(\text{jose}) \leftarrow)$).

Réfuter une hypothèse équivaut à affirmer le contraire ($\text{assert}(\text{president}(\text{jose}))$). Cette hypothèse a été précédemment affirmée par le partenaire. La règle de non-redondance interdit l'usage d'un tel acte.

Dans la situation de jeu 3^{part} , le partenaire dispose de l'argument suivant en faveur de cette hypothèse :

$$A' = (\{\neg\text{president}(\text{jose})\}, \neg\text{president}(\text{jose}) \leftarrow)$$

L'argument A' est non seulement trivial mais sa prémisse est incluse dans le tableau d'engagement. En conséquence, la condition d'énonciation de l'accueil favorable n'est pas vérifiée. D'après l'algorithme 9.5.2, le partenaire répond par une mise en doute.

Réponse à une première mise en doute

Le coup de réponse suivant est émis par le partenaire à l'intention de l'initiateur :

$$\text{move}_4 = \langle \text{part}, \text{init}, \text{move}_3, \text{PP}, \text{challenge}(\neg\text{president}(\text{jose}) \leftarrow) \rangle$$

D'après la règle de « Mise en doute/Réponse », l'initiateur peut répondre par une justification ($\text{assert}(H)$, avec $H \vdash_{\cup} \neg\text{president}(\text{jose}) \leftarrow$), ou par un abandon ($\text{withdraw}(\neg\text{president}(\text{jose}) \leftarrow)$).

Dans la situation de jeu 4^{init} , l'initiateur dispose de l'argument A'_2 qui est faveur de cette hypothèse :

$$A'_2 = (\{r_5, r_{22}(\text{jose})\}, \neg\text{president}(\text{jose}) \leftarrow)$$

Chacun des éléments de la prémisse est soutenu par un argument trivial acceptable. En conséquence la condition d'énonciation ainsi que la tactique d'énonciation de la justification sont vérifiées que l'initiateur soit prévenant ou confiant. Cette hypothèse est justifiée.

Réponse à une troisième affirmation

Le coup de réponse suivant est émis par l'initiateur à l'intention du partenaire :

$$\text{move}_5 = \langle \text{init}, \text{part}, \text{move}_4, \text{PP}, \text{assert}(\{r_5, r_{22}(\text{jose})\}) \rangle$$

Les auditeurs, qu'il s'agisse du partenaire ou de l'agent témoin, mettent à jour le tableau d'engagement correspondant :

$$\begin{aligned} \text{CS}_{\text{init}}^{\text{witness}}(\text{move}_5) &= \text{CS}_{\text{init}}^{\text{witness}}(\text{move}_4) \cup \{r_5, r_{22}(\text{jose})\} \\ \text{CS}_{\text{init}}^{\text{part}}(\text{move}_5) &= \text{CS}_{\text{init}}^{\text{part}}(\text{move}_4) \cup \{r_5, r_{22}(\text{jose})\} \end{aligned}$$

Dans la situation de jeu 5^{part} , le partenaire dispose d'un argument trivial en faveur de chacune de ces deux hypothèses. Contrairement à l'argument trivial en faveur de $r_{22}(\text{jose})$, la prémisse de l'argument trivial en faveur de r_5 est incluse dans le tableau d'engagement. En conséquence, cet ensemble d'hypothèses ne peut être concédé. La règle r_5 est mise en doute.

Réponse à une seconde mise en doute

Le coup de réponse suivant est émis par le partenaire à l'intention de l'initiateur :

$$\text{move}_6 = \langle \text{part}, \text{init}, \text{move}_5, \text{PP}, \text{challenge}(\text{prison}(\text{jose})) \rangle$$

D'après la règle de « Mise en doute/Réponse », l'initiateur peut répondre par une justification ($\text{assert}(H)$, avec $H \vdash_{\mathcal{U}} \text{prison}(\text{jose}) \leftarrow$), ou par un abandon $\text{withdraw}(\text{prison}(\text{jose}) \leftarrow)$.

Dans la situation de jeu 6^{init} , le seul argument en faveur de cette hypothèse dont dispose l'initiateur est l'argument trivial. Parce que la redondance d'hypothèses dans les affirmations d'une même ligne persuasive sont interdites, cette hypothèse est abandonnée.

Clôture du dialogue

Le coup de réponse suivant est émis par l'initiateur à l'intention du partenaire :

$$\text{move}_7 = \langle \text{init}, \text{part}, \text{move}_6, \text{PP}, \text{withdraw}(\text{prison}(\text{jose})) \rangle$$

Ce coup met un terme au dialogue.

Situation finale

Le témoin dispose au terme de la persuasion de trois arguments relatifs au thème. L'argument B' (respectivement A') est un argument trivial en faveur du candidat jose (respectivement en défaveur de ce candidat). L'argument A_2 a été formulé par l'initiateur au cours de la persuasion. Il est en défaveur du candidat jose. L'acceptabilité de ces arguments dépend de la compétence relative des deux joueurs. On distingue deux cas de figure.

Dans le premier cas de figure, le témoin considère l'initiateur plus compétent que le partenaire. Le témoin est représenté dans la figure E.4.0.X. Selon lui, les arguments A' et A_2 sont acceptables. En conséquence, le témoin est partisan du candidat jack.

\ll^*_{witness}	V^*_{witness}	$\mathcal{T}^*_{\text{witness}}$	\mathcal{T}_{Ω_A}
	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	
	v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$	
	$v_{\text{init}}^{\text{witness}}$	$\neg\text{president}(\text{jose}) \leftarrow$ $r_5 : \text{prison}(\text{jose}) \leftarrow$	
	$v_{\text{part}}^{\text{witness}}$	$\text{president}(\text{jose}) \leftarrow$	

Fig. E.4.0.X: Théorie argumentative étendue du témoin au terme de la persuasion qui considère l'initiateur plus compétent que le partenaire

Dans le second cas de figure, le témoin considère l'initiateur moins compétent que le partenaire. Le témoin est représenté dans la figure E.4. Selon lui, l'argument B' est acceptable. En conséquence, le témoin est partisan du candidat jose.

L'avis du témoin au terme de la persuasion dépend de son estimation de la compétence relative des joueurs.

\ll^*_{witness}	V^*_{witness}	$\mathcal{T}^*_{\text{witness}}$	\mathcal{T}_{Ω_A}
	v_1	$r_{11} : \text{president}(\text{jack}) \leftarrow \neg(\text{president}(\text{jose}))$ $r_{21} : \text{president}(\text{jose}) \leftarrow \neg(\text{president}(\text{jack}))$	
	v_2	$r_{12}(x) : \text{prison}(x) \leftarrow \text{escroc}(x)$ $r_{22}(x) : \neg\text{president}(x) \leftarrow \text{prison}(x)$	
	$v_{\text{part}}^{\text{witness}}$	$\text{president}(\text{jose}) \leftarrow$	
	$v_{\text{init}}^{\text{witness}}$	$\neg\text{president}(\text{jose}) \leftarrow$ $r_5 : \text{prison}(\text{jose}) \leftarrow$	

Fig. E.4.0.XI: Théorie argumentative étendue du témoin au terme de la persuasion qui considère l'initiateur moins compétent que le partenaire