Sémantique des phases, réseaux de preuve et divers problèmes de décision en logique linéaire.

Virgile Mogbil

UNIVERSITÉ DE LA MÉDITERRANÉE - AIX-MARSEILLE II
U.F.R. DE SCIENCES

Numéro attribué par la bibliothèque :

THÈSE
pour obtenir le grade de
DOCTEUR DE L'UNIVERSITÉ DE LA MÉDITERRANÉE

Spécialité : MATHÉMATIQUES DISCRÈTES ET
FONDEMENTS DE L'INFORMATIQUE

présentée et soutenue publiquement par

Virgile MOGBIL

le 17/01/2001

TITRE :

SÉMANTIQUE DES PHASES, RÉSEAUX DE PREUVE ET DIVERS PROBLÈMES DE DÉCISION EN LOGIQUE LINÉAIRE

Directeur de thèse :
Yves LAFONT

Rapporteurs :
MM. Michele ABRUSCI
Philippe DE GROOTE

Jury :
MM. Denis BECHET
Nadia CREIGNOU
Philippe DE GROOTE
Jean-Yves GIRARD
Yves LAFONT

A mes parents,
à Hélène et à Gaël pour son premier mois de vie.

Table des matières

Remerciements

Je remercie Yves Lafont qui a suivi et encadré mon travail. Le sujet très interessant qu'il m'a proposé en DEA était un problème qui est toujours ouvert : l'eventuelle décidabilité de MELL. J'ai ainsi pu découvrir la passion qui anime la recherche en la partageant durant toutes ces années. Je le remercie pour sa patience et tout ce temps qu'il m'a consacré.

Je remercie vivement Philippe De Groote et Michele Abrusci de la charge de rapporteur qu'ils ont accepté d'assumer. D'autant plus que les délais ont été assez courts et que notamment le second chapitre traitant de l'accessibilité dans les réseaux de Pétri est difficile et technique.

Je tiens à remercier les membres de l'équipe LDP et tout particulièrement Jean-Yves Girard pour la formidable ambiance au sein de l'équipe, pour les rencontres régulières entre chercheurs de tous pays, pour toutes les idées nouvelles qui ne cessent d'être débattues et pour les repas quotidiens aux discutions mathématiques toujours d'interêt. Merci aussi aux doctorants et post-doctorants de l'équipe pour cette dynamique (et ces rencontres hors de l'IML). Je tiens à remercier tout particulièrement Patrick et Claudia, Pierre, Olivier, Sylvain et Alexandra. Ainsi que Anne, Pierre et Roberto, et tant d'autres. Merci enfin à tout ceux du laboratoire, personnels et chercheurs, et aux personnels et enseignants du departement de mathématiques. Une pensée toute particulière va à René Godet, mon grand-père, qui a su dès mon plus jeune âge éveiller ma curiosité et me passionner pour la recherche (varans dans la baignoire et philosophie !). Merci à mes parents pour tout leur amour. Merci à ma femme qui m'a aidé et soutenu tout ce temps. Et surtout merci pour cet enfant qu'elle vient de me donner.

Introduction

Depuis longtemps les mathématiques sont vues comme un langage universel contenant le sens de ce qu'elles décrivent. Ce réductionnisme de la partie "exacte" des sciences s'est effectué à la fin du $19^{ème}$ siècle à l'aide de systèmes d'inférences. Ainsi le but est d'obtenir toutes les conséquences d'hypothèses quelconques. Cette quête des fondements s'est donc résumée à la recherche d'axiomatisation et de complétude. C'est à dire de systèmes formels complets (i.e. où pour toute formule F on peut prouver F ou sa négation) et dans un tel cadre, de famille d'axiomes permettant de prouver toutes les assertions vraies. Cette quête a aussi porté sur la décision algorithmique, c'est-à-dire la recherche dans l'un des cadres précédents d'algorithmes permettant de décider si une formule est prouvable ou non. Tout ceci se concrétise dans les travaux de G. Frege (1848-1865), de A.N. Whitehead (1861-1947), de B. Russell (1872-1970) et finalement par le programme d'Hilbert. Ce dernier est réfuté par K. Gödel en 1931.

Des théorèmes de complétude ont tout d'abord donné de bons espoirs : axiomatisation complète du calcul propositionnel (E. Post 1921), des logiques du premier ordre (thèse de K. Gödel publiée en 1930). Certains avec en plus des résultats de décidabilité algorithmique comme pour l'arithmétique purement additive (Presburger 1929) et purement multiplicative (Skolem 1930). On montre des résultats de décidabilité du calcul des propositions (Schröder 1890) et de certaines classes préfixielles : Σ_2 (Bernays et Schönfinkel 1928)[1], $\exists^*\forall\forall\exists^*F$ (K. Gödel 1930) qui est la plus large de ces classes décidables.
Enfin les théorèmes d'incomplétude sous différentes formes mettent fin au réductionnisme : l'ensemble des théorèmes dans $(\mathbb{N}\ ;\ =\ ;\ +, \times\ ;\ 0, 1)$ n'est pas récursivement axiomatisable (K. Gödel 1931). C'est-à-dire qu'il existe des énoncés arithmétiques vrais et non prouvables dès que l'on s'intéresse à une famille récursive d'axiomes d'un langage contenant l'addition et la multiplication. On peut étendre ce résultat aux classes de structures permettant de définir l'arithmétique : l'ensemble des théorèmes dans toutes ces structures est indécidable. L'un des corollaire est l'absence de théorème de complétude pour les logiques du second ordre monadique[2].
C'est en 1934 que K. Gödel formalise la notion de récursivité et l'on aboutit avec la thèse de Church (1935) à la réfutation de la décision algorithmique de tels systèmes formels. Enfin c'est en 1936 que A. Turing établit que toute théorie récursivement axiomatisable et complète admet un algorithme de décision, faisant ainsi le lien entre les différentes préoccupations de ce début de siécle.

Les résultats pré-Gödelien sur les réels, les complexes et sur la géométrie

[1] On définit par récurrence les classes $\Sigma_1 = \exists^* F$, $\Pi_1 = \forall^* F$, $\Sigma_{k+1} = \exists^* \Pi_k$ et $\Pi_{k+1} = \forall^* \Sigma_k$ où F est une formule sans quantificateurs.

[2] On définit un langage du second ordre monadique comme un langage du premier ordre avec des variables d'ensembles, du prédicat d'appartenance et des quantifications sur les ensembles.

élémentaire de Tarski (publié en 1948) vont amener de nouvelles illusions. On trouve ainsi de nombreux résultats parmis lesquels :
- la décidabilité de la théorie des groupes abéliens (Szmielew 1949),
- la décidabilité de l'ensemble des formules closes du second ordre monadique réalisables[3] dans certaines structures comme $(\mathbb{N}; =, x \mapsto x+1)$, $(\mathbb{N}; <)$ et $(\mathbb{Z}; <; 0)$ (Büchi 1960-1965), $(\mathbb{Q}; <)$, la famille des parties finies ou dénombrables, la classe des ordres totaux sur un ensemble fini ou dénombrable (Rabin 1969),
- l'établissement de langages maximaux donnant des logiques décidables comme le langage avec égalité, relations unaires, une fonction unaire et des constantes ou le langage sans égalité, relations et fonctions unaires et des constantes. Tout langage contenu dans ces précédents donne une logique décidable,
- des résultats similaires sont obtenus pour les classes préfixielles,

Ainsi bien que les théorèmes d'incomplétude fixe les limites, des résultats de décidabilité n'ont cessé d'affiner les frontières. Ces nouveaux espoirs vont voir leur fin avec les résultats de Meyer (1972) et de Fischer et Rabin (1974) montrant que les algorithmes de cette époque sont de complexité en temps et en espace exponentiels. C'est-à-dire que la réponse au problème de décision ne peut être obtenue de façon raisonnable (cf. paragraphe 1.3). Par la suite les recherches portent sur des résultats de complexité acceptable. Ainsi cette période a vu naître la logique moderne qui n'est que celle des mathématiques : la logique classique parle d'objets statiques qui sont alors vrais ou faux, la logique intuitionniste plus expressive décrit la construction des preuves.

La logique linéaire introduite par J.-Y. Girard en 1987 est basée sur des objets dynamiques. Ils peuvent être traités en logique classique mais au prix de traductions complexes. La prise en compte de ressources, d'états, d'événements, de façon intrinsèque permet qu'elle soit très expressive. Preuve en est des relations profondes qu'elle a avec les concepts informatiques : on peut naturellement simuler les modèles de calculs usuels comme les réseaux de Pétri, les machines de Turing, les machines à registres, etc... Cette expressivité est aussi traduite par le fait que la prouvabilité dans la logique linéaire propositionnelle (i.e. sans quantificateurs) est indécidable.
Quoi de plus naturel alors que de s'intéresser aux problèmes de décision et de complexité en logique linéaire ? Cette thèse développe ce type de sujets :

Le premier chapitre rappelle les définitions du calcul des séquents, des réseaux de preuve et de la sémantique des phases de la logique linéaire. On y retrouve les principaux résultats d'élimination des coupures, de séquentialisation et de critères de correction, de complétude et correction de la sémantique des phases. La seconde partie developpe ce que l'on connaît à propos de la complexité des différents fragments de la logique linéaire et de

[3] On dit qu'une formule est réalisable dans une classe de structure (ou une unique structure) si elle est vraie dans au moins une de ces structures

ses variantes dans le cas propositionnel, quantifié et quand on se limite à ses fragments de Horn. Ce chapitre fini sur quelques considérations sur le fragment multiplicatif exponentiel dont la complexité est toujours inconnue à ce jour.
Le second chapitre traite du problème de l'accessibilité dans les réseaux de Pétri. Quand ce problème n'a pas de réponse positive, on définit un critère semi-linéaire séparant les marquages accessibles de ceux non accessibles. On rappelle l'équivalence entre ce problème d'accessibilité et le problème de décision de la prouvabilité dans le fragment de !-Horn. Cela permet d'établir la complétude de la sémantique des phases semi-linéaires pour ce fragment de la logique linéaire. La complexité du problème de décision pour le fragment multiplicatif exponentiel étant toujours inconnue, une conjecture est d'avoir un résultat de complétude de la sémantique des phases semi-linéaire pour ce fragment. On obtiendrait alors un algorithme de décision.
Le troisième chapitre développe l'idée que l'on peut voir des objets simples additifs comme étant multiplicatifs. On simule, uniquement avec les connecteurs multiplicatifs, le problème des circuits hamiltoniens qui contient l'idée de choix (des arêtes que l'on emprunte) : c'est une notion typiquement additive. Ce codage est ainsi une preuve de la NP-Complétude du fragment multiplicatif de la logique linéaire.
Enfin le dernier chapitre est un travail sur la logique non commutative (NL) qui contient la logique linéaire et la logique linéaire cyclique. On montre un algorithme quadratique permettant d'établir si une structure de preuve de NL est un réseau de preuve de NL. Les précédents algorithmes de ce type sont en temps exponentiel. Ce critère de correction est basé sur la contraction des structures de preuve tout comme l'est celui de V. Danos pour la logique linéaire. A partir de ce dernier S. Guerrini a montré que la correction se fait en temps linéaire. Une conjecture est qu'il en est de même en logique non commutative.

Chapitre 1

Logique linéaire

1.1 Introduction

La logique intuitionniste (LJ) impose d'avoir des séquents ayant au plus une formule à droite. Cette contrainte structurelle fait de LJ une logique constructive, contrairement à la logique classique (LK). En effet, on interdit la duplication de formules (contraction) à droite d'un séquent. L'absence de cette contrainte dans LK assure que l'on ne perd rien à regarder les séquents unilatères (i.e. avec toutes les formules à droite). Le rejet des règles structurelles de contraction et d'affaiblissement (effacement de formules) à droite comme à gauche d'un séquent conduit à la logique linéaire (LL). Ainsi en l'absence de ces règles l'implication linéaire correspond à l'utilisation de la prémisse une fois (i.e. à sa consommation). Pour retrouver la symétrie présente dans LK, on traite la conjonction de façon différente suivant la gestion du contexte : soit il est juxtaposé (conjonction multiplicative) soit il est identifié (conjonction additive). On procède de même pour la disjonction. La négation est involutive aussi on peut décrire les formules de De Morgan sans perdre les propriétés constructives. Pour que cette logique soit expressive et que l'on puisse y traduire les logiques usuelles, on dispose des connecteurs ! et ?, dit exponentiels, qui sont les seuls à permettre des règles logiques d'affaiblissement et de contraction. On obtient par exemple, dans la traduction LJ vers LL, $(A \Longrightarrow B)^0 = !A \multimap B$. En termes de complexité, la preuve d'une formule Γ peut être décrite uniquement à l'aide des sous-formules de Γ. C'est une conséquence de la propriété d'élimination des coupures. De plus dans le cas propositionnel (i.e. sans quantificateurs) le nombre de sous-formules d'une formule donnée est fini. Par règle de contraction, on peut toujours se ramener à des séquents sans répétitions. La recherche de preuve est donc finie. Il existe un algorithme qui détermine s'il existe ou non une telle preuve en un temps fini. Ce problème appelé problème de décision est donc décidable dans LK (c'est un résultat de Cook, qui a montré qu'il est le dual d'un problème NP-complet i.e. co-NP-complet). De même ce problème dans LJ est résoluble par un algorithme non-déterminisme qui utilise un espace polynomial (résultat de Statmann). En logique linéaire, les répétitions sont importantes dans la recherche de preuves depuis une formule : la taille des séquents peut augmenter. Son problème de décision est indécidable comme on le verra par la suite.

1.2 Définitions et résultats

1.2.1 Formules

Les formules de la logique linéaire propositionnelle sont construites à l'aide des constantes et des connecteurs binaires suivants sur les formules atomiques :
– les *multiplicatifs*, $\otimes$ (*tenseur*), $\parr$ (*par*) et leurs constantes 1, $\bot$,

– les *additifs*, $\oplus$ (*plus*), & (*avec*) et leurs constantes 0, $\top$,
et des connecteurs unaires :
– les *exponentiels*, ! (*bien sûr*), ? (*pourquoi pas*),
– les *quantificateurs*, $\forall x$ (*pour tout*), $\exists x$ (*il existe*).
La négation linéaire $(-)^\perp$ est définie pour les formules atomiques positives. Elle est étendue à toutes les formules par les règles suivantes :

$$\begin{array}{ll} A^{\perp\perp} = A & \\ (A \otimes B)^\perp = A^\perp \parr B^\perp & 1^\perp = \perp \\ (A \oplus B)^\perp = A^\perp \,\&\, B^\perp & 0^\perp = \top \\ (!A)^\perp = ?A^\perp & \\ (\forall x\ A)^\perp = \exists x\ A^\perp & \end{array}$$

On note $A \multimap B$ pour $A^\perp \parr B$.

1.2.2 Calcul des séquents

Un *littéral* est une occurrence de formule atomique ou de négation de formule atomique. Un *séquent* $\vdash \Gamma$ est un multi-ensemble d'occurrences de formules que l'on note en général à l'aide de lettres grecques capitales (Γ, Δ). Le calcul des séquents est défini par les groupes de règles suivants :

Groupe identité

$$\frac{}{\vdash A, A^\perp}\ \text{(axiome)} \qquad \frac{\vdash \Gamma, A \quad \vdash A^\perp, \Delta}{\vdash \Gamma, \Delta}\ \text{(coupure)}$$

Groupe logique

multiplicatif

$$\frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}\ \text{(tenseur)} \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\ \text{(par)}$$

$$\frac{}{\vdash 1} \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \perp}$$

additif

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B}\ \text{(plus gauche)} \qquad \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \,\&\, B}\ \text{(avec)}$$

$$\frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B}\ \text{(plus droit)}$$

pas de règle pour 0 $\qquad \dfrac{}{\vdash \top, \Gamma}$

exponentiel

$$\frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}\ \text{(promotion)} \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}\ \text{(déréliction)}$$

$$\frac{\vdash \Gamma}{\vdash \Gamma, ?A}\ \text{(affaiblissement)} \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\ \text{(contraction)}$$

quantificateurs

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, \forall x\ A}\ \text{(pour tout)} \qquad \frac{\vdash \Gamma, A[t/x]}{\vdash \Gamma, \exists x\ A}\ \text{(il existe)}$$

avec x non libre dans Γ

On notera que les règles exponentielles permettent de retrouver le pouvoir expressif des logiques usuelles : il existe des traductions de la logique classique en logique linéaire correctes et complètes pour la prouvabilité.

Théorème 1 (propriété de la sous-formule) *Dans une preuve sans coupure d'un séquent $\vdash A_1, \ldots, A_n$ il n'y a que des sous-formules de $A_1, \ldots, A_n$.*

Théorème 2 (Hauptsatz de Genzen) *Toute preuve d'un séquent se réduit (par élimination des coupures) en une preuve sans coupure.*

Cette réduction termine mais n'est pas confluente : il n'y a pas unicité de la forme normale d'une preuve. C'est une conséquence directe de l'ordre d'introduction des règles dans une preuve. Une autre représentation des démonstrations permet de faire abstraction de cet ordre : les réseaux de preuve.

1.2.3 Réseaux de preuve

On définit un *lien* comme un objet pour lequel les prémisses (arêtes entrantes) et les conclusions (arêtes sortantes) sont deux ensembles disjoints de sommets. Voici les liens utilisés par les réseaux de la logique linéaire multiplicative sans constante : respectivement axiome, coupure, tenseur et par

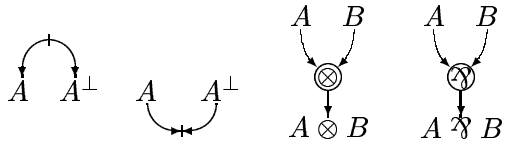


FIG. 1.1 – Liens multiplicatifs

Définition 1 *Une* structure de preuve *G sur les sommets $V(G)$ est un ensemble de liens tel que :*

- *chaque sommet de $V(G)$ est conclusion d'un unique lien,*
- *chaque sommet de $V(G)$ est soit une* conclusion *de G (i.e. n'est prémisse d'aucun lien), soit prémisse d'un unique lien,*
- *l'ensemble des conclusions de G n'est pas vide.*

Les structures de preuves sur un ensemble de sommets étiquetés par des occurrences de formules de la logique linéaire multiplicative sans constante et les symboles "ax" et "cut" (resp. pour "axiome" et "coupure") et dont les liens sont ceux de la figure 1.1, sont appelées structures de preuves multiplicatives.

Définition 2 *Un graphe de correction d'une structure de preuve G est un graphe non orienté $(V(G), \frown)$ tel que :*
- *pour chaque lien binaire sur $s, t \in V(G)$, on a $s \frown t$,*
- *pour chaque lien tenseur sur $s, t, u \in V(G)$ de conclusion u, on a $s \frown u$ et $t \frown u$,*
- *pour chaque lien par sur $s, t, u \in V(G)$ de conclusion u, on a soit $s \frown u$ soit $t \frown u$.*

Définition 3 (Critère de correction de Danos-Regnier) *Une structure de preuve multiplicative est DR-correcte si et seulement si tous ses graphes de correction sont acycliques et connexes. On appelle une telle structure un* réseau multiplicatif.

Théorème 3 (Séquentialisation) *Une structure de preuve (multiplicative) G est la traduction d'une preuve de la logique linéaire (multiplicative) ssi G est un réseau (multiplicatif).*

La propriété d'élimination des coupures pour les réseaux est confluente et fortement normalisable.
La notion de réseaux s'étend à tous les connecteurs de la logique linéaire et l'on conserve la séquentialisation, et la confluence et forte normalisation de l'élimination des coupures.

1.2.4 Sémantique des phases

La logique linéaire a une sémantique des preuves appelée sémantique cohérente (c'est un raffinement des sémantiques de Scott pour la logique intuitionniste) et une sémantique de la prouvabilité, la sémantique des phases que l'on présente ici.

Si $(M, ., 1_M)$ est un monoïde commutatif et $X, Y \subset M$, on pose

$$XY = \{x.y \mid x \in X \text{ et } y \in Y\} \quad \text{et} \quad X \multimap Y = \{z \in M \mid \forall x \in X, x.z \in Y\}$$

Définitions 4 *Un* espace de phases *est un monoïde commutatif M muni d'un sous-ensemble $\perp^\bullet$ de M. Un* fait *est un sous-ensemble X de M tel que $X^{\perp\perp} = X$ où $X^\perp$ désigne $X \multimap \perp^\bullet$.*

En particulier $\bot^\bullet = \{1_M\}^\bot$ est un fait. On pose $1^\bullet = \bot^{\bullet\bot} = \{1_M\}^{\bot\bot}$, $\top^\bullet = M$, $0^\bullet = \top^{\bullet\bot} = \emptyset^{\bot\bot}$ et pour tous faits $X, Y \subset M$,

$$X \otimes Y = (XY)^{\bot\bot} \quad \text{et} \quad X \parr Y = (X^\bot \otimes Y^\bot)^\bot = (X^\bot Y^\bot)^\bot,$$
$$X \,\&\, Y = X \cap Y \quad \text{et} \quad X \oplus Y = (X^\bot \,\&\, Y^\bot)^\bot = (X \cup Y)^{\bot\bot}.$$

Définitions 5 *Un* modèle de phases *est la donnée d'un espace de phase* $(M, \bot^\bullet)$ *et d'une* interprétation *qui associe à chaque atome positif* α *un fait* $\alpha^\bullet$ *de* M. *Un* modèle de phases enrichi *est un modèle de phases muni d'un sous-monoïde* K *de* $J(M) = \{x \in 1^\bullet \mid x \in \{x^2\}^{\bot\bot}\}$.

L'interprétation des modèles est étendue aux formules par induction à l'aide des règles précédentes et des règles :

$$?X = (X^\bot \cap K)^\bot \quad \text{et} \quad !X = (X \cap K)^{\bot\bot}.$$

Définitions 6 *On dit que le modèle* M satisfait *la formule* A *si* $1_M \in A^\bullet$, *on le note* $M \models A$.

Le modèle syntaxique $LL^\bullet$ est constitué du monoïde commutatif libre engendré par les formules de la logique linéaire. Le produit $\Gamma.\Delta$ sur les séquents est Γ, Δ et l'élément neutre est le séquent vide. $\bot^\bullet$ est l'ensemble des séquents prouvables sans coupures et K engendré par les formules de type $?A$. L'interprétation d'une formule atomique positive est $\alpha^\bullet = \{\alpha\}^\bot$.

Lemme 4 (Okada) *Pour toute formule* A, *on a* $A^\bot \subseteq \{A\}^\bot$ *dans* $LL^\bullet$.

Ainsi toute formule satisfaite dans $LL^\bullet$ est prouvable sans coupure.

Théorème 5 (Correction, complétude et élimination des coupures) *Pour toute formule* A, *on a :*

$$\vdash A \iff \forall (M, \bot^\bullet),\ M \models A \iff LL^\bullet \models A \iff \vdash A \text{ sans coupure.}$$

On peut de même montrer la correction et la complétude de la sémantique des phases finiment engendrée. C'est-à-dire engendrée par les sous-formules d'une formule donnée. On obtient ainsi "gratuitement" la propriété de la sous-formule. On a cette propriété de complétude pour tous les fragments de la logique linéaire.
Si l'on se restreint au fragment multiplicatif et additif de la logique linéaire alors on peut établir ce type de résultat pour la sémantique des phases finie [Laf97] car on a un nombre fini de faits. C'est ce que l'on appelle la propriété des modèles finis. Comme conséquence, on a la décidabilité de la prouvabilité dans ce fragment. On n'a pas la complétude de la sémantique des phases finie pour la logique linéaire entière, ni pour son fragment sans les additifs. Ceci sera traité à la fin de ce chapitre.

Sans aller aussi loin, on peut se servir d'arguments de sémantique des phases pour établir une condition nécessaire à la prouvabilité d'une formule. On obtient alors un contre-modèle. En voici une illustration : le fait qu'une formule F du fragment multiplicatif est équilibrée[1] est une telle condition, le modèle de phases $(\mathbb{Z}, +)$ sur l'ensemble des atomes de F avec $\bot = \{0\}$ muni de l'interprétation $a^\bullet = \{1\}$ pour tout atome a définit un contre modèle pour toute formule non équilibrée (par exemple $(a \multimap a \otimes a)^\bullet = \{1\}$). On verra aussi par la suite que la sémantique des phases peut-être utile pour prouver que le codage d'un problème en logique linéaire est fidèle.

1.2.5 Variations et généralisations

Par définition, l'échange est implicite dans les séquents. On peut aussi présenter un calcul des séquents avec une règle d'échange explicite où les séquents ne sont plus des multi-ensembles :

$$\frac{\vdash \Gamma}{\vdash \Delta} \text{ avec } \Delta \text{ permutation de } \Gamma$$

Cette règle assure la commutativité du tenseur, ainsi si l'on considère le calcul sans cette règle d'échange, on obtient la logique linéaire non commutative : la question du choix d'une ou deux négations se pose. Les solutions les plus élégantes sont obtenues avec une seule négation, ce qui a pour conséquence l'obtention de preuves inchangées par permutation circulaire : c'est la logique linéaire cyclique. On peut ainsi voir le calcul de Lambek, introduit pour traiter des questions de linguistique, comme les prémices de la logique linéaire. La règle d'échange se traduit en sémantique des phases par la contrainte $\forall x, y, z \in M\ xyz \in \bot \Longrightarrow xzy \in \bot$. La cyclicité correspond à $\forall x, y \in M\ xy \in \bot \Longrightarrow yx \in \bot$. On verra au dernier chapitre la logique non commutative (NL) qui généralise la logique linéaire et la logique linéaire cyclique.

Les autres règles structurelles présentes dans la logique classique et intuitionniste, peuvent être considérées. Si on ajoute la règle d'affaiblissement (non logique), c'est-à-dire que l'on peut effacer une formule non exponentielle d'un séquent, on obtient la logique linéaire affine :

$$\frac{\vdash \Gamma}{\vdash \Gamma, A} \text{ (affaiblissement structurel)}$$

Cette règle se généralise en sémantique des phases par la contrainte suivante sur $\bot$: $\forall x \in M\ x \in \bot \Longrightarrow \forall y \in M\ xy \in \bot$. On verra par la suite les conséquences de ces variations sur l'expressivité des différents fragments à travers leur complexité.

[1] Chaque atome apparaît autant de fois positivement que négativement, une définition précise dans le cas intuitionniste est donnée au paragraphe 3.7.

Enfin on peut ajouter la règle de contraction structurelle : on se permet la duplication de formules non exponentielles. Ce qui se traduit en sémantique des phases par la contrainte $\forall x, y \in M\ x.y^2 \in \perp \Longrightarrow xy \in \perp$. Combinée à la règle d'affaiblissement structurel, on retrouve l'échange structurel et donc la logique classique (on perd tout bénéfice).

1.3 Complexité des fragments de la logique linéaire

Comme on l'a vu au paragraphe 1.2.2, les règles logiques de la logique linéaire se décomposent en différents groupes : les multiplicatifs (M) avec les connecteurs $\otimes$, $\parr$ et les constantes 1 et $\top$, les additifs (A) avec les connecteurs $\oplus$, $\&$ et les constantes 0 et $\perp$, les exponentiels (E) ! et ?, et les quantificateurs. On désigne ainsi de façon équivalente la logique linéaire propositionnelle par la notation LL ou MAELL. On fera de même pour les fragments de la logique linéaire c'est-à-dire les formules construites seulement sur certains groupes : MLL désigne la logique linéaire propositionnelle multiplicative, MALL la logique linéaire propositionnelle multiplicative et additive, etc... On note LL1 pour la logique linéaire quantifiée au premier ordre, LL2 pour le second ordre et LW pour la logique linéaire affine c'est-à-dire avec la règle structurelle d'affaiblissement. La restriction d'un fragment à sa partie intuitionniste, c'est-à-dire à des séquents ayant au plus une formule à droite, est désigné par ILL, ILL1 ou ILL2 suivant le cas. Enfin on note CyLL pour la logique linéaire cyclique et NL pour la logique non commutative qui généralise LL et CyLL.

1.3.1 Notations, rappels et définitions

Définition 7 *Soient f et g des fonctions de $\mathbb{N}$ dans $\mathbb{N}$. On définit les classes de fonctions asymptotiques suivantes :*

$$O(f(n)) = \{g(n) \mid \textit{ il existe } c, n_0 \textit{ entiers positifs } n > n_0 \Longrightarrow c.f(n) \geqslant g(n)\}$$

$$\Omega(f(n)) = \{g(n) \mid \textit{ il existe } c, n_0 \textit{ entiers positifs } n > n_0 \Longrightarrow c.f(n) \leqslant g(n)\}$$

$$\Theta(f(n)) = O(f(n)) \cap \Omega(f(n))$$

Une fonction de $\Theta(f(n))$ est donc de l'ordre de grandeur de f. C'est-à-dire qu'elle a le même comportement asymptotique. Par abus de notations, on utilise $O(f(n)), \Omega(f(n))$ et $\Theta(f(n)$ pour désigner le représentant d'une classe et $g(n) = O(f(n))$ pour signifier que $g(n) \in O(f(n))$.

Définition 8 *On appelle classe de complexité $TIME(f(n))$ (respectivement $SPACE(f(n))$) l'ensemble des langages reconnus en temps f (resp. en espace f) par une machine de Turing. De même l'ensemble des langages reconnus par les machines de Turing non déterministes en un temps f définit la classe de complexité $NTIME(f(n))$.*

Définition 9 *On appelle P (resp. EXP) l'union des classes de complexité $TIME(n^k)$ (resp. $TIME(2^{n^k})$) et NP (resp. $NEXP$) l'union des classes de complexité $NTIME(n^k)$ (resp. $NTIME(2^{n^k})$). On appelle $PSPACE$ l'union des classes de complexité $SPACE(n^k)$.*

On a les résultats suivants :

$$P \subseteq NP \subseteq PSPACE \subseteq EXP \subseteq NEXP.$$

1.3.2 Calcul propositionnel sans quantificateurs

Le problème de la prouvabilité dans LL est indécidable. Ce résultat est obtenu en codant le problème indécidable de l'arêt pour une certaine forme de machine à registres : les machines à registres et embranchements (ACM) mais sans test à zéro (car il n'est pas naturellement codable dans LL). On se sert des modalités pour pouvoir réutiliser des instructions, des additifs pour gérer les embranchements et des multiplicatifs pour coder les valeurs dans les registres. Voici une version simplifiée du résultat originel de [LMSS92].

Plus formellement, une ACM est la donnée d'un ensemble fini d'états dont certains sont distingués (l'état initial Q_I et au moins un état final Q_F) et un ensemble fini de transitions. On se limite à une ACM à deux registres. Une configuration de la machine est une séquence de triplets (Q_i, x, y) constitués de l'état courant et des valeurs x et y des deux registres A et B. L'ensemble des transitions possibles est de la forme suivante :
- $(Q_i + A\ Q_j)$ passe de la configuration (Q_i, x, y) à $(Q_j, x+1, y)$,
- $(Q_i - A\ Q_j)$ passe de la configuration (Q_i, x, y) à $(Q_j, x-1, y)$ si A est non nul,
- $(Q_i + B\ Q_j)$ et $(Q_i - B\ Q_j)$ font de même pour le second registre,
- $(Q_i fork\ Q_j\ Q_k)$ passe de la configuration (Q_i, x, y) à $((Q_j, x, y), (Q_k, x, y))$ (i.e. à deux exécutions en parallèle).

On dit que la configuration initiale est acceptée si et seulement si l'exécution de la machine finie dans chaque branche par une configuration où l'état est l'état final et où les valeurs des deux registres sont nulles.

Ainsi on code une telle machine dans LL à l'aide d'un ensemble d'atomes $\{q_i \mid Q_i\}_i \cup \{q_I, q_F\} \cup \{a, b\}$ et l'on se sert d'un tenseur de n littéraux pour coder le fait qu'un registre contienne la valeur n. Une transition $(Q_i + A\ Q_j)$ est codée par la formule $(q_i \multimap q_j \otimes a)$, une transition $(Q_i - A\ Q_j)$ par $(q_i \otimes a \multimap q_j)$ et une transition $(Q_i fork\ Q_j\ Q_k)$ par $(q_i \multimap (q_j \,\&\, q_k))$. Pour finir une configuration (Q_i, x, y) est codée par le séquent $\vdash !\Theta, q_i, a^x, b^y, q_F^\perp$ où Θ est l'ensemble des formules codant les transitions de la machine et q_F correspond à l'état final.

Le théorème suivant permet d'établir le résultat recherché : soit une ACM donnée, la configuration initiale est acceptée si et seulement si le séquent correspondant est dérivable dans LL.

MELL

Le problème de décision de la prouvabilité pour MELL est l'un des seuls restant encore ouverts. La conjecture d'Yves Lafont est qu'il est décidable. Le chapitre suivant est une première étape vers sa preuve éventuelle. Une discussion préliminaire sur ce fragment se trouve à la fin de ce chapitre.

MALL

Voici une idée de preuve de la PSPACE-Complétude du problème de décision pour MALL : on s'aperçoit que dans une preuve sans coupure de MALL, les hypothèses comportent moins de symboles que les conclusions (seules les exponentielles permettent la réutilisation de certaines formules). Si l'on cherche la preuve d'une formule dans le calcul des séquent en la devinant de bas en haut, on a alors une borne linéaire en la profondeur de la preuve sans coupures de MALL. Comme ce fragment satisfait l'élimination des coupures, il existe ainsi un algorithme PSPACE (non déterministe) pour décider de la prouvabilité d'un séquent de MALL.

Pour établir que ce fragment est PSPACE-difficile, on peut coder les formules booléennes quantifiées de la logique classique [LMSS92]. On se sert des additifs pour coder les quantificateurs, $\forall x$ par $x \,\&\, x^\perp$ et $\exists x$ par $x \oplus x^\perp$, et des multiplicatifs pour s'assurer de leur bon comportement (mise sous forme prénexe).

MLL

Si comme précédemment, on cherche à deviner la preuve d'une formule de MLL depuis le bas alors on peut montrer que l'on analyse exactement une fois chaque connecteur dans une preuve sans coupure. Le problème de décision pour MLL est donc dans NP.

La preuve originale de NP-Complétude de ce fragment utilise le codage du problème des trois partitions [Kan92]. On verra au chapitre 3 une autre preuve de ce résultat par le codage d'un problème de théorie des graphes, classique en complexité : le problème des circuits hamiltoniens.

autres fragments

Etonnamment, on retrouve toute la puissance expressive quand on restreint les précédents fragments aux éléments neutres (pas de littéraux). Certains de ces résultats peuvent être obtenus depuis ceux de [LMSS92], mais on préférera les preuves plus simples de [Kan95, LW94].

Si l'on se restreint aux fragments intuitionnistes (c'est-à-dire aux séquents avec au plus une formule à droite), on ne change généralement pas la complexité du problème de décision.

Certaines restrictions affines comme MALW (corollaire des résultats de MALL) et MLW [LMSS92] conservent la même complexité que leurs fragments non affaiblis. Mais ce n'est pas le cas de tous : la décidabilité de MAELW fait intervenir un codage du problème d'équipartition [Kop95]. On notera qu'il existe une preuve simple en sémantique des phases affines (on impose à $\bot^\bullet$ d'être un idéal dans le monoïde) qui implique la propriété des modèles finis [Laf97]. De même, si l'on se limite aux éléments neutres, tous les fragments affines ont un problème de décision linéaire.

Enfin le cas des fragments cycliques, CyLL et MECyLL sont indécidables. Il suffit de simuler une machine à deux registres : l'état (Q_i, x, y) est codé par le séquent $\vdash a^x, q_i, b^y, !\Theta$ où les atomes codent les états et la valeur des registres comme dans la preuve d'indécidabilité de LL. Les transitions sont codées par des exponentielles : l'exécution d'une transition est une duplication et une commutation jusqu'au lieu d'application, où l'on change l'état courant et la valeur d'un registre. La preuve originale code les machines à registres par une réduction aux systèmes semi-Thue [LMSS92].

1.3.3 Premier ordre et ordres supérieurs

Les fragments entiers et les fragments multiplicatifs exponentiels au premier et au second ordre de la logique linéaire, de la logique affine et de la logique linéaire cyclique sont indécidable [Gir87] (LL1, LW1, CyLL1, MELL1, MELW1, MECyLL1, LL2, LW2, CyLL2, MELL2, MELW2, MECyLL2). Ces résultats sont obtenus pour la plupart à l'aide de translation des logiques du premier et second ordre, classique et intuitionniste ou comme corollaires de ces résultats. Le tableau suivant résume ce qu'il en est pour les autres fragments :

Fragments		Complexité
MA	LL1 et LW1	NEXP-Complet [LS94a, LS94b]
M	LL1 et LW1	NP-Complet [LS94a, Kan94a]
MA et M	LL2	Indécidable [Laf96, LS96, LSS95]
MA et M	LW2	Indécidable [Kop95]
MA et M	CyLL2	Indécidable

On remarquera notamment quelques techniques comme dans la preuve de l'indécidabilité de MALL2 [Laf96] qui utilise une approche similaire à une preuve pour LL (simulation de machine à registres) encodant la contraction par les quantificateurs $(\forall\alpha(\alpha \,\&\, 1) \multimap \alpha \otimes \alpha)$ et la déréliction et l'affaiblissement par les additifs $(\theta \,\&\, 1)$. La preuve de l'indécidabilité de MLL2 utilise quant à elle la sémantique des phases pour s'assurer de la fidélité du codage. Les preuves de NEXP-complétude et NP-complétude utilisent un codage de la quantification universelle et d'unification pour instantier les quantificateurs existentiels.

On sait aussi que MACyLL1 est dans NEXP [LS94b] et que MCyLL1 est dans NP [LS94b].

1.3.4 Fragments de Horn

Les plus petits fragments de la logique propositionnelle sont le fragment de Horn et ses généralisations additives et exponentielles. On considère ici des implications, dites de Horn, de la forme $X \multimap Y$ où X et Y sont des tenseurs de littéraux. On étend cette notion aux implications $\oplus$-Horn de la forme $X \multimap (Y_1 \oplus Y_2)$ et &-Horn de la forme $((X_1 \multimap Y_1) \,\&\, (X_2 \multimap Y_2))$. On définit ainsi des séquents, dit de Horn, de la forme $G, W \vdash Z$ où G est un multi-ensemble d'implications de Horn et W, Z des tenseurs de littéraux. On étend de même cette notion aux séquents de $\oplus$-Horn, de &-Horn et de $(\oplus, \&)$-Horn. Enfin on généralise tout ces types de séquents par ceux de la forme $!G, W \vdash Z$ où $!G$ désigne le multi-ensemble formé des $!F$ pour chaque formule F de G. On nomme ces derniers !-Horn, $(!, \oplus)$-Horn, etc... On note HLL pour les fragments de Horn de la logique linéaire. Voici un résumé de leurs différentes complexités que l'on peut trouver en grande partie dans les articles de M. Kanovich [Kan92, Kan94a, Kan94b, Kan94c, Kan96] :

Fragments	Complexité
HLL, $\oplus$-HLL, &-HLL	NP-Complet
$(\oplus, \&)$-HLL	PSACE-Complet
$(!, \oplus)$-HLL	Indécidable[2]
!-HLL, $(!, \&)$-HLL	Décidable

[2] Voir la preuve de l'indécidabilité de LL du paragraphe 1.3.2.

Les résultats de NP-complétude sont obtenus par équivalence avec l'existence de programmes de Horn (graphes orientés acycliques étiquetés par des implications de Horn munis d'une notion d'exécution correspondant à une preuve). Ils codent le problème NP-complet des trois partitions. Il est intéressant de remarquer que le problème de décision en logique intuitionniste est PSACE-complet alors que pour son fragment de Horn, il est décidable en temps linéaire. On constate que ce n'est pas du tout le cas des fragments de Horn de la logique linéaire qui, bien qu'ils soient les fragments les plus simples, contiennent pour la plupart déjà tout le pouvoir expressif.

On notera que les problèmes de décision respectifs pour les séquents de !-Horn de la logique linéaire et pour ceux de $(!, \&)$-Horn sont exactement équivalents au problème d'accessibilité dans les réseaux de Pétri. On en donne une preuve dans le chapitre 2.

Si l'on s'intéresse aux fragments de Horn de la logique linéaire affine, on remarque que $\oplus$-HLW est PSACE-Complet. Ce résultat est obtenu par un codage du fragment implicatif de la logique intuitionniste. Il sert aussi à

montrer que $(\oplus, \&)$-HLL est de même complexité. L'ajout de la règle d'affaiblissement ne change rien pour les autres fragments de Horn.

1.3.5 Quelques mots sur MELL

Le fragment multiplicatif de la logique linéaire peut être présenté comme une formalisation logique des mécanismes de base des calculs parallèles. Les réseaux de Pétri qui sont l'un des premiers modèles de systèmes parallèles et l'un des plus utilisés entrent exactement dans ce formalisme. On montre ainsi que toute exécution correspond à une preuve et réciproquement [Asp87, GG89, MOM91].
Informellement, les ressources dans un réseau de Pétri sont représentées par des *places* qui peuvent contenir un nombre arbitraire de jetons. On peut les voir comme un multi-ensemble auquel on fait correspondre un tenseur d'atomes identiques ayant la même multiplicité (trois jetons dans la place A correspondent à la ressource $a \otimes a \otimes a$). L'ensemble des jetons à un instant donné est appelé *marquage*. Les *transitions* du réseau sont des opérations qui consomment des ressources de certaines places et en produisent d'autres dans d'autres places. On ne peut les exécuter que si l'on dispose des ressources nécessaires. On les voit comme des implications linéaires du tenseurs d'atomes des jetons consommés vers le tenseur d'atomes des jetons produits (la transition qui prend deux jetons de la place A pour produire un jeton dans les places B et C correspond à la formule $a \otimes a \multimap b \otimes c$). La question de savoir si pour un réseau de Pétri et deux marquages M et M' donnés, il existe une suite de transitions exécutables qui produit M' depuis M est appelée problème d'accessibilité dans les réseaux de Pétri. Elle correspond exactement à la prouvabilité du séquent de !-Horn associé (une preuve formelle est détaillée dans le chapitre 2). Le fragment utilisant les connecteurs !, $\multimap$ et un seul littéral, est le plus petit fragment de MELL permettant de coder ce problème. Il existe des résultats de correction et d'adéquation des réseaux de Pétri pour d'autres fragments de LL. C'est-à-dire que les réseaux de Pétri en sont des modèles [EW90, EW93a, EW93b].
Le problème de l'accessibilité dans les réseaux de Pétri est décidable (on le sait EXPSPACE-difficile [Lip76]). Une preuve de la décidabilité de MELL en serait une pour ce problème. On ne connaît pas de codage des formules de MELL dans les réseaux de Pétri.

Sémantique des phases pour MELL

On a vu au paragraphe 1.2.4 que la sémantique des phases finies est complète pour MLL et donc MLL satisfait la propriété des modèles finis. Ce résultat n'est plus vrai pour MELL [Laf97] : la formule $\varphi = !a \otimes !(a \otimes b) \otimes !(a \otimes b \multimap 1) \multimap b$ où a et b sont des atomes positifs est satisfaite dans tout modèle de phases finies. En effet dans un tel modèle, le nombre de faits est fini donc

il existe $p < q$ tel que $(b^p)^\bullet = (b^q)^\bullet$. D'où $b^p \multimap b^q$ est satisfait. Or à partir $!(a \otimes b)$ on peut prouver $a^p \otimes b^p$, donc la formule $[!(a \otimes b)] \multimap (a^p \otimes b^q)$ est satisfaite, c'est-à-dire $!(a \otimes b) \models a^p \otimes b^q$. De même les assertions suivantes sont vraies :
- $(a^p \otimes b^q) \otimes (!a) \models a^{q-1} \otimes b^q$,
- $(a^{q-1} \otimes b^q) \otimes [!(a \otimes b \multimap 1)] \models b$.

Donc la formule φ est satisfaite, cependant elle n'est pas prouvable !
On remarquera que la formule φ est associée à un reseau de Pétri.
Le résultat du chapitre 2 est une première étape dans l'espoir d'établir un résultat plus faible que la complétude de la sémantique des phases finies pour MELL mais qui permet de garder encore un contrôle sur les faits : la sémantique des phases semi-linéaires est complète pour !-HLL. La conjecture est que cela reste vrai pour MELL. On pourrait alors encore énumérer et calculer, et ainsi déterminer un algorithme de décision.

Problèmes équivalents au problème de désicion pour MELL

On peut ramener le problème de décision pour le fragment intuitionniste formé sur les connecteurs $\otimes, \multimap, 1, !$ (IMELL) à celui pour le fragment dont les séquent sont $!\Theta, \Gamma \vdash A$ où toutes les formules sont de la forme $a \otimes b \multimap c$, $a \multimap b \otimes c$, $a \multimap 1$, $1 \multimap a$, $a \multimap (b \multimap c)$ ou $(a \multimap b) \multimap c$ avec a, b, c atomes.
On remarquera que dans ce fragment, il n'y a pas de formule comportant des sous-formules exponentielles autre qu'elle même. Pour cela on décompose chaque formule de IMELL algorithmiquement. Pour une formule de IMELL donnée F, on introduit un nouvel atome pour chacune de ses sous-formules. Soient A et B les sous-formules de F et soient p et q les nouveaux atomes leurs correspondants. On construit récursivement l'ensemble Θ sur les sous-formules de F :
- si $F = A \otimes B$ est codée r alors on ajoute à Θ les formules $r \multimap p \otimes q$ et $p \otimes q \multimap r$,
- si $F = A \multimap B$ est codée s alors on ajoute à Θ les formules $(s \otimes p) \multimap q$ et $(p \multimap q) \multimap s$,
- si $F = !A$ est codée t alors on ajoute à Θ les formules $t \multimap p$ (pour simuler la déréliction), $t \multimap 1$ (pour simuler l'affaiblissement) et $t \multimap t \otimes t$ (pour simuler la contraction).

On remarquera que seule la formule $(p \multimap q) \multimap s$ ne correspond pas à un codage des réseaux de Pétri.
Il ne manque dans cette reformulation simple du fragment IMELL que la simulation de la règle de promotion. Pour ce faire, on sature l'ensemble des formules prouvables à partir de certaines formules. Soit Φ l'ensemble des atomes codant les sous-formules exponentielles (i.e. les "t"). Soit $\Gamma \subseteq \Phi$ un multi-ensemble sans répétitions. Si $!\Theta, \Gamma \vdash p$ est prouvable (où p est le nom d'une sous-formule A de F) alors on ajoute à Θ la formule $\Gamma \multimap t$ où t est le nom de la sous-formule $!A$. Ainsi on simule la règle

$$\frac{!\Theta, \Gamma \vdash p}{!\Theta, \Gamma \vdash t}.$$

On réitére ce procédé pour chaque sous-formule de F (i.e. pour chaque p) et pour chaque sous-ensemble sans répétitions de Φ. L'ensemble Θ sert ainsi à simuler les règles du calcul des séquents nécéssaires à la preuve d'une formule donnée de MELL. Comme le cardinal de Φ est fini, cette procédure termine.

On obtient ainsi l'équivalence entre les problèmes de décision pour MELL et pour le fragment formé des séquents $\vdash ?A_1, \ldots, ?A_n, \Lambda$ où les A_i et Λ sont dans MLL. De cette façon les A_i sont les formules de Θ. Si on note $A^{(n)}$ la formule composée par le tenseur de n formules A alors décider la prouvabilité d'une formule de MELL est équivalent à l'existence d'entiers $n_1, \ldots, n_k$ tels que $\vdash A_1^{(n_1)}, \ldots, A_k^{(n_k)}, \Lambda$.

1.4 Logique linéaire élémentaire et allégée

La logique linéaire allégée (LLL) [Gir98] permet de capturer la classe des fonctions calculables en temps polynomial. On borne a priori le temps de normalisation : une borne est donnée sur le nombre d'étapes nécessaires à la construction des preuves sans coupures en fonction de la formule conclusion et de la profondeur d'une preuve (c'est-à-dire le nombre de ! imbriqués nécessaires). Cette dernière implique une modification du calcul des séquents. Les principes fondateurs de LLL sont :

$$!(A \,\&\, B) \simeq !A \otimes !B \qquad !A \multimap ?A.$$

On restreint la modalité ! à la fonctorialité (depuis $A \vdash B$ on déduit $!A \vdash !B$). Si on a toujours la contraction et l'affaiblissement, on ne permet plus la déréliction ni le *digging* ($!A \vdash !!A$). On se sert donc d'une nouvelle modalité § qui vérifie $!A \multimap \S A$ et $\S A \otimes \S B \multimap \S(A \otimes B)$. On obtient alors le théorème d'expressivité suivant :

Théorème 6 *Si $f : \mathbb{N} \longrightarrow \mathbb{N}$ est calculable en temps polynomial alors il existe une preuve de LLL (de conclusion $(!^k 1 \otimes bin) \multimap \S^k bin$) qui la représente (c'est-à-dire qui la calcule).*

Pour ELL on ajoute :

$$!A \otimes !B \simeq !(A \otimes B).$$

La modalité ! est alors multi-fonctoriel et la modalité § n'est plus nécessaire. ELL permet de caractériser la classe des fonctions élémentaires de Kalmar [DJ99].

Bibliographie

[Asp87] A. Asperti. A logic for concurrency. Technical report, Dipartemento di informatica, Universitá di pisa, 1987.

[DJ99] V. Danos and J.-B. Joinet. Linear logic and elementary time. In *First workshop on implicit computational complexity (ICC'99)*, 1999.

[EW90] U. Engberg and G. Winskel. Petri nets as models of linear logic. In A. Arnold, editor, *Proceedings CAAP '90*, volume 431 of *Lecture Notes in Computer Science*, pages 147–161, Copenhagen, 1990. Springer-Verlag.

[EW93a] U. Engberg and G. Winskel. Completeness results for linear logic on Petri nets. In A. Borzyszkowski and S. Sokolowski, editors, *Proceedings of MFCS'93 (Mathematical Foundations of Computer Science), Gdańsk, Poland, August 1993*, volume 711 of *Lecture Notes in Computer Science*, pages 442–452. Springer-Verlag, 1993.

[EW93b] U. Engberg and G. Winskel. Linear logic on petri nets. *Lecture Notes in Computer Science*, 803 :176–229, 1993.

[GG89] C.A. Gunter and V. Gehlot. Nets as tensor theories. In G. De Michelis, editor, *Proc. 10-th International Conference on Application and Theory of Petri Nets, Bonn*, 1989.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50, 1987.

[Gir98] J.-Y. Girard. Light linear logic. *Information and Computation*, 143 :175–204, 1998.

[Kan92] Max I. Kanovich. Horn programming in linear logic is NP-complete. In *Proceedings 7th Annual IEEE Symposium on Logic in Computer Science, LICS'92, Santa Cruz, CA, 22–25 June 1992*, pages 200–210. IEEE Computer Society Press, Los Alamitos, California, 1992.

[Kan94a] Max I. Kanovich. The complexity of Horn fragments of linear logic. *Annals of Pure and Applied Logic*, 69 :195–241, 1994.

[Kan94b] Max I. Kanovich. Linear logic as a logic of computations. *Annals of Pure and Applied Logic*, 67(1-3) :183–212, 1994. Also in *Logic at Tver '92*, Sokal, Russia, July 1992.

[Kan94c] Max I. Kanovich. Petri nets, Horn programs, linear logic, and vector games. In M. Hagiya and J. Mitchell, editors, *Proceedings of the International Symposium Theoretical Aspects of Computer Software TACS'94*, volume 789 of *Lecture Notes in Computer Science*, pages 642–666, Sendai, Japan, 1994. Springer-Verlag.

[Kan95] Max I. Kanovich. The complexity of neutrals in linear logic. In D. Kozen, editor, *Tenth Annual IEEE Symposium on Logic in Computer Science*, pages 486–495, San Diego, California, jun 1995.

[Kan96] Max I. Kanovich. Linear logic automata. *Annals of Pure and Applied Logic*, 78 :147–148, 1996.

[Kop95] A. P. Kopylov. Decidability of linear affine logic. In D. Kozen, editor, *Tenth Annual IEEE Symposium on Logic in Computer Science*, pages 496–504, San Diego, California, june 1995.

[Laf96] Y. Lafont. The undecidability of second order linear logic without exponentials. *Journal of Symbolic Logic*, 61 :541–548, 1996.

[Laf97] Y. Lafont. The finite model property for various fragments of linear logic. *Journal of Symbolic Logic*, 62(4) :1202–1208, 1997.

[Lip76] R. Lipton. The reachability problem is exponential-space-hard. Technical Report 62, Dept. Comp. Science, Yale Univ., New Haven, CT, 1976.

[LMSS92] P. Lincoln, J. Mitchell, A. Scedrov, and N. Shankar. Decision problems for propositional linear logic. *Annals of Pure and Applied Logic*, 56 :239–311, 1992. Also in the Proceedings of the 31th Annual Symposium on Foundations of Computer Science, St Louis, Missouri, October 1990, IEEE Computer Society Press. Also available as Technical Report SRI-CSL-90-08 from SRI International, Computer Science Laboratory.

[LS94a] P. Lincoln and A. Scedrov. First order linear logic without modalities is NEXPTIME-hard. *Theoretical Computer Science*, 135(1) :139–154, 1994.

[LS94b] P. Lincoln and N. Shankar. Proof search in first order linear logic and other cut-free sequent calculi. In *Proceedings of the Ninth Annual Symposium on Logic in Computer Science, Paris, France*, pages 282–291. IEEE Computer Society Press, 1994.

[LS96] Y. Lafont and A. Scedrov. The undecidability of second order multiplicative linear logic. *Information and Computation*, 125(1) :46–51, 1996.

[LSS95] P. Lincoln, A. Scedrov, and N. Shankar. Decision problems for second order linear logic. In *Proceedings of the 10-th Annual IEEE Symposium on Logic in Computer Science, San Diego, California*, Los Alamitos, California, 1995. IEEE Computer Society Press.

[LW94] P. Lincoln and T. Winkler. Constant-only multiplicative linear logic is NP-complete. *Theoretical Computer Science*, 135(1) :155–169, 1994.

[MOM91] N. Martí-Oliet and J. Meseguer. From Petri nets to linear logic. *Mathematical Structures in Computer Science*, 1 :66–101, 1991. Revised version of paper in LNCS 389 (1989).

Chapitre 2

Sémantique des phases semi-linéaire

2.1 Introduction

Si l'on résume quelques idées que l'on vient de voir, on s'aperçoit que le problème de décision de MLL est NP-complet, que celui de LL est indécidable et que la complexité de celui de MELL est inconnue. Cependant on peut coder dans ce dernier fragment le problème de l'accessibilité dans les réseaux de Pétri qui est EXPSPACE-difficile[Lip76]. De plus MELL est exactement réductible à des séquents qui ne comportent que des exponentielles en "surface" et des formules "simples" où seules celles de la forme $(p \multimap q) \multimap s$ ne correspondent pas directement à un codage dans les réseaux de Pétri (cf. paragraphe 1.3.5). Une approche sémantique de l'expressivité de MELL montre qu'il ne vérifie pas la propriété des modèles finis (cf. paragraphe 1.3.5). Yves Lafont conjecture que la sémantique des phases semi-linéaire est complète pour ce fragment. L'espoir qu'une étude détaillée de la preuve difficile de l'accessibilité dans les réseaux de Pétri permette de mieux comprendre le fragment MELL a motivé le travail de ce chapitre. On sait entre autre que l'ensemble des marquages accessibles depuis un marquage donné n'est pas semi-linéaire. Le résultat essentiel de ce chapitre est que dans le cas d'une réponse négative au problème d'accessibilité, on peut définir un ensemble semi-linéaire séparant les marquages accessibles depuis le marquage initial, du marquage final. On obtient alors comme corollaire que la sémantique des phases semi-linéaire est complète pour le fragment de !-Horn (dont le problème de décision est exactement équivalent à celui de l'accessibilité dans les réseaux de Pétri). Voici l'organisation de ce chapitre :

La première partie définit les réseaux de Pétri introduits par A.C. Pétri en 1962 et les systèmes d'addition de vecteurs qui sont des modèles équivalents. Les problèmes classiques d'accessibilité y sont énoncés.

La seconde partie introduit la notion de critère, utilisée dans le théorème principal, ainsi qu'un corollaire établissant la complétude de la sémantique des phases semi-linéaire pour le fragment de !-Horn. Il est rappelé tout ce qui est nécéssaire à sa compréhension et à sa preuve.

La troisième partie concerne des résultats démontrés par S.R. Kosaraju [Kos82] sur la décidabilité du problème d'accessibilité dans les réseaux de Pétri. On pourra se référer à l'ouvrage de C. Reutenauer [Reu89] pour en comprendre tous les détails. Ce problème remonte à 1969 (Karp et Miller) et a eu deux démonstrations complètes indépendantes par Mayr [May81, May84] et par Kosaraju (1982).

Ces résultats et plus particulièrement les parties constructives des preuves sont nécessaires à la dernière partie où l'on prouve toujours de façon constructive le théorème principal de ce chapitre.

2.2 Problèmes d'accessibilité

2.2.1 Réseaux de Pétri et Systèmes d'addition de vecteurs

Définition 10 *Un* réseau de Pétri *est un quadruplet* $R = (P, T, Pré, Post)$ *où*
- P *est un ensemble fini d'éléments appelés* places *;*
- T *est un ensemble fini d'éléments appelés* transitions *;*
- $Pré$ *est une application* $P \times T \longrightarrow \mathbb{N}$ *appelée l'application d'*incidence avant *;*
- $Post$ *est une application* $T \times P \longrightarrow \mathbb{N}$ *appelée l'application d'*incidence arrière.

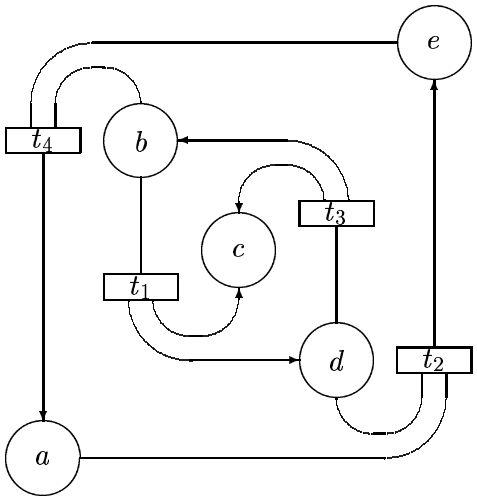


FIG. 2.1 – Un exemple de réseau de Pétri

Définition 11 *Un* système d'addition de vecteurs *est la donnée d'un entier* m *et d'un ensemble fini* V *de vecteurs dans* $\mathbb{Z}^m$.

Exemple 1 $m = 5$ et $V = \{v_1, v_2, v_3, v_4\}$ où

$$\begin{array}{lllllllllll} v_1 = (& 0 & , & -1 & , & 1 & , & 1 & , & 0 &) \\ v_2 = (& -1 & , & 0 & , & 0 & , & -1 & , & 1 &) \\ v_3 = (& 0 & , & 1 & , & 1 & , & -1 & , & 0 &) \\ v_4 = (& 1 & , & -1 & , & 0 & , & 0 & , & -1 &) \end{array}$$

définissent un système d'addition de vecteurs correspondant au réseau de Pétri de la figure 2.1. Les vecteurs correspondent aux transitions et les coordonnées aux places (la première coordonnée à la place a, etc).

2.2.2 La $\mathbb{Z}$-accessibilité

Définitions 12 *Un* ensemble linéaire $L \subseteq \mathbb{N}^m$ *est un ensemble de la forme* $L = c + \sum_{i=1}^{k} \mathbb{N}p_i$ *avec* $k \in \mathbb{N}$. *Le vecteur* $c \in \mathbb{N}^m$ *est appelé* base *de* L *et les vecteurs* $p_1, \ldots, p_k \in \mathbb{N}^m$ *sont appelés* périodes *de* L. *On le notera aussi* $L = (c; p_1, \ldots, p_k)$.
Un ensemble semi-linéaire *est une union finie d'ensembles linéaires.*

Définitions 13 *Etant donnés un système d'addition de vecteurs* $S = (m, V)$ *et deux vecteurs* M, M' *dans* $\mathbb{Z}^m$, *on dit que* M' *est* $\mathbb{Z}$-accessible *depuis* M *s'il existe une suite de vecteurs de* V $s = v_0, \ldots, v_n$ *telle que* $M' = M + \sum_{i=0}^{n} v_i$. *On le note* $M \xrightarrow{s} M'$.

Le problème de la $\mathbb{Z}$-accessibilité *dans les systèmes d'addition de vecteurs est, étant donnés un système d'addition de vecteurs* $S = (m, V)$ *et deux vecteurs* M, M' *dans* $\mathbb{Z}^m$ *de décider s'il existe une suite* s *de vecteurs de* V *telle que* $M \xrightarrow{s} M'$.

Exemple 2 Si $n \in \mathbb{N}^*$ alors $M' = (0, 0, 2n, 0, 0)$ est $\mathbb{Z}$-accessible depuis $M = (0, 0, 0, 0, 0)$ dans le système d'addition de vecteurs de l'exemple 1 car $M' = M + n.v_1 + n.v_3$. On remarquera que quelque soit la combinaison de ces vecteurs, on ne peut obtenir M' depuis M en restant dans $\mathbb{N}^m$.

Théorème 7 *Le problème de la* $\mathbb{Z}$*-accessibilité dans les systèmes d'addition de vecteurs est décidable.*

Preuve. Voir [Reu89] (ou [Mog95] pour une preuve directe). □

2.2.3 La $\mathbb{N}$-accessibilité

Définition 14 *un* marquage *du réseau de Pétri* $R = (P, T, Pré, Post)$ *est une application* $M : P \to \mathbb{N}$. *On identifiera souvent un marquage avec le vecteur qu'il définit dans* $\mathbb{N}^P$.

Définitions 15 *Etant donnés un réseau de Pétri* R *et un marquage* M, *on dit que la transition* t *est* franchissable *pour* M *si l'on a pour tout* $p \in P$ $M(p) \leqslant Pré(p, t)$.
Soit t *franchissable pour* M *et soit* M' *le marquage définit par* $M'(p) = M(p) - Pré(p, t) + Post(t, p)$, *on dit que* M' *est obtenu depuis* M *par* franchissement *de* t, *ce que l'on note* $M(t > M'$.
Etant donnés un réseau de Pétri R, *une suite* $s = (t_1, \ldots, t_n)$ *de transitions et un marquage* M, *on dit que cette suite est* franchissable *pour* M *s'il existe une suite de marquages* $M_0, \ldots, M_n$ *avec* $M_0 = M$ *telle que pour chaque indice* $i \in \{1, \ldots n\}$, *on ait :* $M_{i-1}(t_i > M_i$. *On dit alors que* M_n *est* $\mathbb{N}$-accessible *depuis* M *et on le note* $M(s > M_n$.

Le problème de la $\mathbb{N}$-accessibilité *dans les réseaux de Pétri est, étant donnés un réseau de Pétri R et deux marquages M, M', de décider s'il existe une suite s de transitions franchissable pour M et telle que $M(s > M'$.*

Exemple 3 Dans le réseau de Pétri de la figure 2.1, la transition t_1 est franchissable pour le marquage $(0,1,0,0,0)$ et le marquage obtenu est $(0,0,1,1,0)$. Pour tout n entier naturel (pair), le marquage $(1,0,n(n+1)/2,0,0)$ est $\mathbb{N}$-accessible depuis $M_0 = (1,n,0,0,0)$ par une suite de transitions du type $(t_1^* t_2 (t_3^* t_4)^*)^*$. Si on ajoute une première place permettant d'atteindre une valeur quelconque en seconde coordonnée puis ensuite une transition initiant le précédent réseau de Pétri alors l'ensemble des marquages accessibles depuis $(1,0,0,0,0)$ donné n'est pas semi-linéaire. On a cependant le résultat suivant :

Théorème 8 *Le problème de la $\mathbb{N}$-accessibilité dans les réseaux de Pétri est décidable.*

Preuve. Voir [Kos82] et [Reu89]. □

2.3 Existence de critères semi-linéaires et corollaire

On énonce le théorème principal de ce chapitre : c'est un résultat d'existence dont la preuve est constructive. Un corollaire de ce théorème établit un résultat de complétude.

2.3.1 Notion de critère

Définitions 16 *Un* critère *du système d'addition de vecteurs $S = (m, V)$ est un ensemble de points de $\mathbb{N}^m$ tels que pour tout point x du critère, pour tout $v_i \in V$, si $x + v_i \in \mathbb{N}^m$ alors $x + v_i$ est un point du critère.*
Un critère *du réseau de Pétri $R = (P, T, Pré, Post)$ est un ensemble de marquages clos par $(t >$ pour tout $t \in T$.*

Il est clair que ces deux notions sont équivalentes :
- si x est un point du critère et si $v_i \in V$ tel que $x + v_i \in \mathbb{N}^m$ alors il existe dans le réseau de Pétri correspondant une transition t franchissable pour le marquage x tel que $x(t > x'$ et $x' = x + v_i$.
- si M est un marquage du réseau de Pétri et si $t \in T$ tel que $M(t > M'$ alors il existe dans le système d'addition de vecteurs correspondant un vecteur $v \in \mathbb{Z}^m$ tel que $M + v = M'$.

Théorème 9 *Etant donnés un réseau de Pétri R et deux marquages M, M', si M' n'est pas accessible depuis M alors il existe un critère semi-linéaire $\mathcal{A}$ pour le réseau de Pétri R tel que $M \in \mathcal{A}$ et $M' \notin \mathcal{A}$.*

La preuve de ce théorème fait l'objet de la suite de ce chapitre (sections 2.4 et 2.5).

2.3.2 Fragments de Horn et sémantique de phases semi-linéaire

Définitions 17 *Une* conjonction simple *est un tenseur de littéraux positifs. Une* implication de Horn *est un séquent de la forme $X \multimap Y$ où X et Y sont des conjonctions simples.*
On appelle fragment de !-Horn *le fragment de la logique linéaire formé des séquents*

$$!\Gamma, W \vdash Z$$

où

W, Z sont des conjonctions simples,
$!\Gamma$ est formé des formules $!\gamma$ pour chaque $\gamma \in \Gamma$ où
Γ est un multi-ensemble composé uniquement d'implications de Horn.

Lemme 10 *Le problème de décision pour le fragment de !-Horn est exactement équivalent au problème de la $\mathbb{N}$-accessibilité dans les réseaux de Pétri.*

Preuve. Si $R = (P, T, Pré, Post)$ est un réseau de Pétri ayant n transitions et m places et M, M' sont deux marquages de $\mathbb{N}^m$ alors on définit les formules suivantes sur les atomes $a_1, \ldots, a_m$:

$$W = \bigotimes_{i=1}^{m} a_i^{(M)_i},\ Z = \bigotimes_{i=1}^{m} a_i^{(M')_i} \text{ et}$$

$$G = \left\{ X_i \multimap Y_i \mid X_i = \bigotimes_{i=1}^{m} a_i^{Pré(t_i)} \text{ et } Y_i = \bigotimes_{i=1}^{m} a_i^{Post(t_i)} \text{ pour } t_i \in T \right\}$$

Alors M' est $\mathbb{N}$-accessible depuis M dans R si et seulement si $!G, W \vdash Z$. Réciproquement si $!G, W \vdash Z$ est un sequent de !-Horn ayant m atomes positifs distincts $a_1, \ldots, a_m$ avec

$$W = \bigotimes_{i=1}^{m} a_i^{\alpha_i},\ Z = \bigotimes_{i=1}^{m} a_i^{\beta_i} \text{ et}$$

$$G = \{X_i \multimap Y_i\}_{i \in H} \text{ où l'on note } X_i = \bigotimes_{i=1}^{m} a_i^{x_i} \text{ et } Y_i = \bigotimes_{i=1}^{m} a_i^{y_i}$$

alors on définit le réseau de Pétri $R = (P, T, Pré, Post)$ et les marquages M, M' par :

$$n = Card(H),\ n' = \sum_{j \in H'} n_j \text{ et } n + n' = Card(T)$$

$$\forall i \in \{1, \ldots, n\}\ Pré(t_i) = x_i \text{ et } Post(t_i) = y_i \text{ pour } t_i \in T,$$

$$\forall i \in \{1, \ldots, m\}\ (M)_i = \alpha_i \text{ et } (M')_i = \beta_i.$$

Alors M' est $\mathbb{N}$-accessible depuis M dans R si et seulement si $!G, W \vdash Z$. □

Définitions 18 *Un* modèle de phases semi-linéaire *est un modèle de phases tel que* $\perp^\bullet$ *est semi-linéaire.*

Lemme 11 *Soit* $(M, \perp^\bullet)$ *un espace de phases et* $X, Y \subset M$.
i) $X \subset X^{\perp\perp}$,
ii) Si $X \subset Y$ *alors* $Y^\perp \subset X^\perp$,
iii) $(X^{\perp\perp}Y^{\perp\perp})^{\perp\perp} = (XY)^{\perp\perp}$.

Preuve. Soit $(M, \perp^\bullet)$ un espace de phases et $X, Y \subset M$.
i) Par définition $X^\perp = \{\overline{x} \in M \mid \forall x \in X\ \overline{x}x \in \perp^\bullet\}$ donc si $x \in X$ alors $\forall \overline{x} \in X^\perp$ on a $\overline{x}x \in \perp^\bullet$ i.e. $x \in X^{\perp\perp}$.
ii) Par définition $\overline{y} \in Y^\perp$ si $\forall y \in Y\ \overline{y}y \in \perp^\bullet$ donc si $X \subset Y$ alors $\forall x \in X\ \overline{y}x \in \perp^\bullet$ i.e. $\overline{y} \in Y^\perp$.
iii) Montrons que $(XY)^{\perp\perp} \subset (X^{\perp\perp}Y^{\perp\perp})^{\perp\perp}$:
D'après *i)* on a $XY \subset X^{\perp\perp}Y^{\perp\perp}$ donc d'après *ii)* on a le résultat.
Montrons que $(X^{\perp\perp}Y^{\perp\perp})^{\perp\perp} \subset (XY)^{\perp\perp}$:
Si $A, B \subset M$ alors $AB \subset \perp^\bullet \longleftrightarrow A \subset B^\perp$. Donc on a

$$\begin{aligned}
(XY)^\perp(X^{\perp\perp}Y^{\perp\perp}) \subset \perp^\bullet &\longleftrightarrow (XY)^\perp X^{\perp\perp} \subset (Y^{\perp\perp})^\perp = Y^\perp \\
&\longleftrightarrow (XY)^\perp X^{\perp\perp}Y \subset \perp^\bullet \\
&\longleftrightarrow (XY)^\perp Y \subset (X^{\perp\perp})^\perp = X^\perp \\
&\longleftrightarrow (XY)^\perp XY \subset \perp^\bullet \\
&\longleftrightarrow (XY)^\perp \subset (XY)^\perp
\end{aligned}$$

□

Définition 19 *On dit qu'une sémantique* S *est complète pour le fragment de* !*-Horn si, pour tout modèle* M *de* S $M \models A$, *implique* $\vdash A$ *dans ce fragment.*

Corollaire 12 *La sémantique des phases semi-linéaire est complète pour le fragment de* !*-Horn.*

Preuve. Soit $!G, W \vdash Z$ un séquent de !-Horn ayant m atomes positifs distincts $a_1, \ldots, a_m$. Soient le réseau de Pétri R et les marquages M, M' correspondants (cf. preuve du lemme 10).
Soit $(\mathcal{M}, \perp^\bullet, K)$ le modèle de phases semi-linéaire enrichi où l'on a $(\mathcal{M}, \cdot) = (\mathbb{N}^m, +)$, $\perp^\bullet = \mathcal{A}^\complement$ et $K = \{\overline{0}\}$ où $1_\mathcal{M} = (0, \ldots, 0) = \overline{0}$ et $\mathcal{A}^\complement$ désigne le complémentaire de $\mathcal{A}$: l'ensemble $\perp^\bullet$ est stable par rétro-transitions.
On associe aux atomes l'interprétation $a_i^\bullet = \{e_i\}^{\perp\perp}$ où $(e_i)_j = 1$ si $j = i$ et $(e_i)_j = 0$ sinon.
Comme $\overline{0} \in \perp^{\bullet\perp} = 1^\bullet$ car $\forall b \in \perp^\bullet\ b + \overline{0} \in \perp^\bullet$ et comme $\overline{0} \in \{\overline{0} + \overline{0}\}^{\perp\perp}$ car $\overline{0} = \overline{0} + \overline{0}$ et $X \subset X^{\perp\perp}$, on a bien $K \subset J(\mathcal{M})$.

Si on note $W = \otimes_{i\in I} a_i^{\alpha_i}$ alors d'après le lemme 11 on a

$$W^\bullet = \left\{ \sum_{i\in I} \sum_{j=1}^{\alpha_i} a_i^\bullet \right\}^{\perp\perp} = \left\{ \sum_{i\in I} \sum_{j=1}^{\alpha_i} e_i \right\}^{\perp\perp} = \{M\}^{\perp\perp},$$

de même

$$Z^\bullet = \{M'\}^{\perp\perp}.$$

Le modèle satisfait les "transitions". En effet pour tout $F \in G$, $1_\mathcal{M} \in F^\bullet$ par définiton de $\perp^\bullet$. Donc $1_\mathcal{M} \in (!F)^\bullet$ car $1_\mathcal{M} \in X$ implique $1_\mathcal{M} \in !X$.
Par définition $!X = (X \cap K)^{\perp\perp}$ or $K = \{\overline{0}\}$ d'où $!X = \{\overline{0}\}^{\perp\perp}$ pour tout fait X, ainsi

$$1_\mathcal{M} \in (!G)^\bullet$$

Montrons que si $!G, W \vdash Z$ et $M' \in \perp^\bullet$ alors $M \in \perp^\bullet$.
Or $!G, W \vdash Z$, donc $W^\bullet \subset Z^\bullet$ c'est à dire $\{M\}^{\perp\perp} \subset \{M'\}^{\perp\perp}$ donc an a $\{M'\}^\perp \subset \{M'\}^\perp$.
Or $\{v\}^\perp = \{z \mid vz \in \mathcal{A}^\complement\}$
donc si $M' \in \perp^\bullet$ alors $1_\mathcal{M} \subset \{M\}^\perp$ c'est à dire $M \in \perp^\bullet$. □

Remarque Les précédents résultats restent vrai avec le fragment de $(!, \&)$-Horn constitué des séquents de la forme $!\Gamma, W \vdash Z$ où W, Z sont des conjonctions simples et Γ est un multi-ensemble de formules de Horn et de &-Horn. C'est à dire respectivement de la forme $X_1 \multimap Y_1$ et $(X_1 \multimap Y_1)\&(X_2 \multimap Y_2)$ avec X_1, X_2, Y_1, Y_2 des conjonctions simples.

2.4 Les résultats intermédiaires connus

On trouve ici un résumé de résultats obtenus par S.R.Kosaraju. Quelques affinements mineurs ont été faits pour obtenir une preuve directe des lemmes établissant le théorème principal de ce chapitre. Seuls les résultats nécéssaires à la compréhension et les parties de leurs preuves utiles au théorème final sont indiqués.
On généralise donc les systèmes d'addition de vecteurs en chaîne et on définit toute la terminologie associée. On étudie des constructions sur les chaînes permettant d'établir diverses propriétés du système. Enfin on définit un algorithme de décidabilité fonctionnant par réduction de chaînes.

2.4.1 Généralisation des systèmes

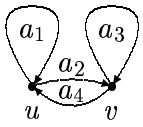
Système d'addition de vecteurs avec états (SAVE)

Un SAVE est un système d'addition de vecteurs où l'addition des vecteurs est contrôlée par un automate ayant un nombre fini d'états. De façon plus formelle,

Définitions 20 *Un* SAVE *est la donnée d'un graphe orienté fini* $G = (Q, A)$, *d'un entier* $m \geqslant 1$ *et d'une application* $v : A \to \mathbb{Z}^m$. Q *est l'ensemble des* états, A *l'ensemble des* arcs *et* $v(a)$ *est la* valuation *ou* étiquette *de l'arc* a. *Une* configuration *de* G *est un couple* (q, y) *formé d'un état de* G *et d'un point de* $\mathbb{N}^m$.

On voit facilement que la notion de SAVE généralise celle de système d'addition de vecteurs : un système $S = (m, V)$ est simplement un SAVE à un seul état p dont les boucles $p \to p$ sont étiquetées par les vecteurs de V.

Exemple 4 Le SAVE suivant correspond au système d'addition de vecteurs de l'exemple 1 où l'on a modélisé des coordonnées par des états :



avec $\begin{cases} v(a_1) = (\ -1\ ,\ 1\ ,\ 1\ ) \\ v(a_2) = (\ -1\ ,\ 0\ ,\ 0\ ) \\ v(a_3) = (\ 1\ ,\ 1\ ,\ -1\ ) \\ v(a_4) = (\ 0\ ,\ 0\ ,\ -1\ ) \end{cases}$

Chaîne de systèmes d'addition de vecteurs avec états

Une chaîne de systèmes d'addition de vecteurs avec états (on dira plus simplement chaîne) est un SAVE particulier muni de contraintes sur certains de ses états. Ce SAVE est sous la forme d'une suite de SAVE reliés entre eux par un arc allant de l'état final à l'état initial. Ce sont ces états de la chaîne qui subissent les contraintes. On représente une chaîne à l'aide du schéma suivant :

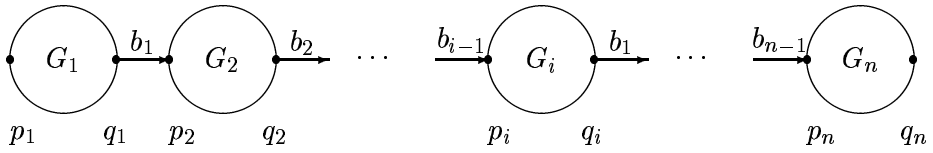


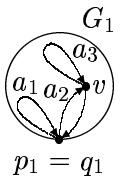
FIG. 2.2 – Une chaîne

Définitions 21 *Etant donné un SAVE étiqueté dans* $\mathbb{Z}^m$ *et un état* q, *on appelle* contrainte sur q *un vecteur* f *dans* $(\mathbb{N} \cup \{\omega\})^m$ *où* ω *est un nouveau symbole dont le sens intuitif est "indéterminé".*
On dit qu'un vecteur $x \in \mathbb{Z}^m$ vérifie la contrainte f *si pour tout* $i \in \{1, \ldots, m\}$, *on a soit* $f_i = \omega$ *et* $x_i \geqslant 0$, *soit* $f_i = x_i$. *On notera que* x *est alors dans* $\mathbb{N}^m$.
Une chaîne est la donnée d'une suite de SAVE $G_1, \ldots, G_n$, *d'une suite d'arcs* $b_1, \ldots, b_{n-1}$ *et d'un ensemble de contraintes* $\{e_1, s_1, e_2, s_2, \ldots, e_n, s_n\}$ *telle que pour chaque* $G_i = (Q_i, A_i)$ *on a distingué des états* p_i, q_i *où* q_i *est relié à* p_{i+1} *par l'arc* b_i *pour* $i \neq n$, e_i *est la contrainte sur* p_i *et* s_i *est la contrainte sur* q_i.

On appelle e_i (respectivement s_i) la contrainte d'entrée *de G_i (respectivement* de sortie*) et p_i (respectivement q_i) l'*état d'entrée *de G_i (respectivement* de sortie*).*
On note $E_i = \{j \in \{1, \dots, m\} \mid (e_i)_j \neq \omega\}$ l'ensemble des coordonnées contraintes à l'entrée *de G_i et $S_i = \{j \in \{1, \dots, m\} \mid (s_i)_j \neq \omega\}$ l'ensemble des* coordonnées contraintes à la sortie *de G_i. On note $R_i = \{j \in \{1, \dots, m\} \mid \forall a \in A_i \ (v(a))_j = 0\}$ l'ensemble des* coordonnées rigides *de G_i.*

Un SAVE est donc une chaîne ayant pour seule contrainte d'entrée et de sortie le vecteur $(\omega, \dots, \omega)$.

Exemple 5
Soit C la chaîne formée d'un seul SAVE (i.e. $n = 1$) : celui de l'exemple 4 avec $p_1 = q_1 = u$. On a $m = 3$, soient $e_1 = (\omega, 0, 0)$ la contrainte d'entrée de G_1 et $s_1 = (0, \omega, 0)$ la contrainte de sortie de G_1. Donc $E_1 = \{2, 3\}$, $S_1 = \{1, 3\}$ et $R_1 = \emptyset$.



Définitions 22 *Etant donné un chemin $c : p_1 \to q_n$ dans une chaîne, il existe une unique décomposition de c de la forme*

$$c_1 b_1 c_2 b_2 \dots b_{n-1} c_n$$

où chaque c_i est un chemin dans G_i.
La promenade $P = (c, x)$ associée à ce chemin définit pour chaque i, le point d'entrée *de P dans G_i par $x_i = x + v(c_1 b_1 \dots b_{i-1})$ et le* point de sortie *de P dans G_i par $y_i = x + v(c_1 b_1 \dots b_{i-1} c_i)$.*
Les points intermédiaires dans *G_i de P sont les points intermédiaires de la promenade (c_i, x_i).*
Une promenade est positive *si pour chaque $i \in \{1, \dots, n\}$ ses points intermédiaires dans G_i sont dans $\mathbb{N}^m$.*
Une promenade est contrainte *si son état initial est p_1, son état final est q_n et si pour chaque $i \in \{1, \dots, n\}$ x_i vérifie la contrainte e_i et y_i vérifie la contrainte s_i.*
Pour $x \in \mathbb{N}^m$ on notera $(p_1, x) \longrightarrow (q_n, y)$ si la promenade $P = (p_1, x)$ est contrainte et $(p_1, x) \longrightarrow^+ (q_n, y)$ si elle est contrainte et positive. Pour $J \subset \{1, \dots, m\}$ on notera de même $(p_1, x) \longrightarrow_J^+ (q_n, y)$ si elle est contrainte et positive relativement à J i.e. si l'on se restreint aux coordonnées dans J.

Le problème de la $\mathbb{N}$-accessibilité *pour des vecteurs $x, y \in \mathbb{N}^m$ donnés et une chaîne ayant pour contrainte d'entrée x et de sortie y, est de savoir s'il existe une promenade contrainte et positive.*

Exemple 6 Soit $c = a_1 a_1 a_2 a_3 a_4$ un chemin dans la chaîne C de l'exemple 5. La promenade $P = (c, (2, 0, 0))$ a pour point d'entrée dans G_1 le vecteur $x_1 = (2, 0, 0)$ et pour point de sortie dans G_1 le vecteur $y_1 = x_1 + v(a_1) +$

$v(a_1)+v(a_2)+v(a_3)+v(a_4) = (0,3,0)$. Les configurations intermédiaires de P sont :
$(p_1,(2,0,0))$,$(p_1,(1,1,1))$,$(p_1,(0,2,2))$, $(v,(0,2,1))$,$(v,(1,3,0))$,$(q_1,(0,3,0))$.
Or x_1 vérifie la contrainte d'entrée $e_1 = (\omega,0,0)$ et y_1 vérifie la contrainte de sortie $s_1 = (0,\omega,0)$ donc P est une promenade contrainte et positive de la chaîne C.

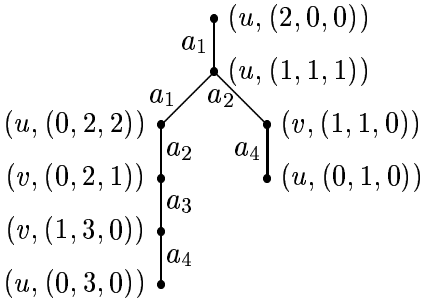
2.4.2 Arbre de Karp et Miller, graphe de couverture

Soit un SAVE dont le graphe sous-jacent est $G = (Q,A)$ et la valuation $v : A \longrightarrow \mathbb{Z}^m$. Soit une configuration initiale $(p,x) \in Q \times \mathbb{N}^m$.

L'*arbre de Karp et Miller* T associé au SAVE et à la configuration initiale (p,x) est un arbre dont les sommets sont étiquetés dans $Q \times (\mathbb{Z} \cup \{\infty\})^m$ et les arcs étiquetés par un arc de G. Sa racine est étiquetée (p,x) et on le définit récursivement sur ces sommets déjà construit de la façon suivante : soit s un sommet de T étiquetée (q,y),

SI (q,y) est l'étiquette d'un ancêtre de s ALORS s n'a pas de fils,
SINON s a un fils $s(a)$ pour chaque arc $a = (q,r)$ de G tel que $y + v(a) \geq 0$. L'arc est étiqueté par a et le sommet $s(a)$ est étiqueté par (r,z) où z est définit sur ses coordonnées par $\forall i \in \{1,\ldots,m\}$:
s'il existe un ancêtre de $s(a)$ dont l'étiquette est (r,z') avec $z' \leq y + v(a)$ et $z'_i < (y+v(a))_i$ alors $z_i = \infty$.
sinon $z_i = (y+v(a))_i$.

Exemple 7 Voici l'arbre de Karp et Miller sur le SAVE de l'exemple 4, depuis la configuration initiale $(u,(2,0,0))$:



Le *graphe de couverture* d'un SAVE s'obtient en confondant dans l'arbre de Karp et Miller tout sommet terminal ayant un ancêtre avec la même étiquette, avec cet ancêtre.

Théorème 13 *Etant donné un SAVE fortement connexe, un état p, un vecteur $x \in \mathbb{N}^m$, un ensemble de coordonnées $J \subset \{1,\ldots,m\}$, la propriété suivante est décidable :*

Il existe un vecteur $\Delta \in \mathbb{Z}^m$ *tel que* $\Delta_J \geq (1, \ldots, 1)$ *et que* $(p, x + \Delta)$ *soit* $\mathbb{N}$*-accessible à partir de* (p, x) *par rapport à* J.

De plus, si cette propriété est fausse, on peut effectivement déterminer un entier N *tel que : pour tout chemin* c *du SAVE d'origine* p *et admissible pour* x *par rapport à* J*, il existe* $j \in J$ *tel que pour tout point intermédiaire* z *de la promenade* (c, x) *on ait* $z_j \leq N$.

Remarque La dernière partie de ce théorème est effectivement calculable à l'aide du graphe de couverture du SAVE.

2.4.3 La propriété θ

Soit C une chaîne de longueur n et un vecteur $x \in \mathbb{N}^m$.
On dit que C possède la *propriété* $\theta 1$ si pour tout entier naturel N il existe une promenade contrainte $P = (c, x)$ telle que
i) Pour tout arc a dans $A_1 \cup \cdots \cup A_n$, le chemin c utilise au moins N fois l'arc a.
ii) Pour tout $i \in \{1, \ldots, n\}$, toute coordonnée non contrainte en entrée du point d'entrée de P dans G_i est $\geqslant N$.
On dit que C possède la *propriété* $\theta 2$ si
i) Pour tout $i \in \{1, \ldots, n\}$, il existe $\Gamma_i \in \mathbb{Z}^m$ tel que
   – si $j \in E_i \backslash R_i$ alors $(\Gamma_i)_j \geqslant 1$,
   – $(p_i, e_i') \longrightarrow^+_{E_i \backslash R_i} (p_i, e_i' + \Gamma_i)$ dans G_i.
ii) Pour tout $i \in \{1, \ldots, n\}$, il existe $\Delta_i \in \mathbb{Z}^m$ tel que
   – si $j \in S_i \backslash R_i$ alors $(\Delta_i)_j \geqslant 1$,
   – $(q_i, s_i' + \Delta_i) \longrightarrow^+_{S_i \backslash R_i} (q_i, s_i')$ dans G_i.
où $e_i' \in \mathbb{N}^m$ (resp. s_i') coïncide avec e_i (resp. s_i) sur ses coordonnées $\neq \omega$ et est nul ailleurs.

L'intêret de cette propriété réside dans les deux résultats suivant :

Théorème 14 *On peut décider si une chaîne donnée possède la propriété* $\theta = \theta_1 \wedge \theta_2$.

Théorème 15 *Si une chaîne donnée possède la propriété* θ*, alors il existe une promenade positive et contrainte.*

2.4.4 Opérations sur les chaînes

Taille d'une chaîne

Définitions 23 *on muni* $\mathbb{N}^3$ *de l'*ordre lexicographique*, c'est à dire de l'ordre suivant :*

$(a, b, c) \leqslant (a', b', c')$ *si* $(a < a')$ *ou* $(a = a'$ *et* $b < b')$ *ou* $(a = a'$ *et* $b = b'$ *et* $c \leqslant c')$

Si $(A, \leqslant)$ est un ensemble ordonné alors on dit que l'ordre est arnitien *s'il n'existe pas de suite infinie strictement décroissante dans A.*
On remarquera que l'ordre lexicographique est arnitien sur $\mathbb{N}^3$.
Soit $(\mathbb{N}^3)^\star$ le monoïde libre engendré par $\mathbb{N}^3$, i.e. l'ensemble des mots sur $\mathbb{N}^3$, sur lequel on définit l'ordre suivant :

$$U \prec V \text{ si } \begin{cases} \text{il existe } X, Y, W \in (\mathbb{N}^3)^\star \text{ avec } W \neq \text{ mot vide,} \\ \text{il existe } u \in \mathbb{N}^3 \text{ tels que } U = XuY \text{ et } V = XWY \\ \text{et pour toute lettre } v \text{ de } W,\ u \geqslant v \end{cases}$$

Lemme 16 *La relation définie par $U \leqslant V$ si on a la cloture réflexive et transitive de $\prec$ entre U et V est un ordre arnitien sur $(\mathbb{N}^3)^\star$.*

Définition 24 *On appelle* taille *d'une chaîne C la suite finie de vecteurs de $\mathbb{N}^3$*

$$\mid C \mid = (\mid G_1 \mid, \ldots, \mid G_n \mid) \text{ où } \mid G_i \mid = (m- \mid R_i \mid, \mid A_i \mid, 2m- \mid E_i \mid - \mid S_i \mid).$$

La comparaison de tailles est faite avec l'ordre sur $(\mathbb{N}^3)^\star$ défini au lemme 16.

Remarque La finitude de l'algorithme de Kosaraju provient du caractère artinien de cet ordre sur les tailles de chaînes.

Déploiement selon un arc

Lemme 17 *Etant donné une chaîne C, un arc a et un entier N, il existe une chaîne C' telle qu'il y ait bijection entre les promenades positives et contraintes de C dont le chemin associé utilise exactement N fois l'arc a et les promenades positives et contraintes de C'. De plus la taille de C' est inférieure à celle de C.*

Soit i l'indice du SAVE auquel appartient l'arc $a = (u, v)$. On définit la chaîne C' comme une copie de C dans laquelle on remplace le SAVE $G_i = (Q_i, A_i)$ par $N + 1$ copies de $G'_i = (Q_i, A_i \backslash \{a\})$ reliées entre elles par l'arc a. On schématise cette réduction par :
La première copie de G'_i a pour contrainte d'entrée e_i, celle des autres est $(\omega, \ldots, \omega)$. La dernière copie de G'_i a pour contrainte de sortie s_i, celle des autres est $(\omega, \ldots, \omega)$.
On dira que l'on a déployé C selon l'arc a à l'ordre k.

Rigidification selon un SAVE

Lemme 18 *Etant donné une chaîne C, un indice de SAVE i, un indice de coordonnée j et un entier N, il existe un ensemble de chaînes $\{C_{j,k}\}_{(j,k)\in K}$ telles qu'il y ait bijection entre les promenades positives et contraintes de C dont les points intermédiaires dans G_i ont la j-ième coordonnée $\leqslant N$ et la*

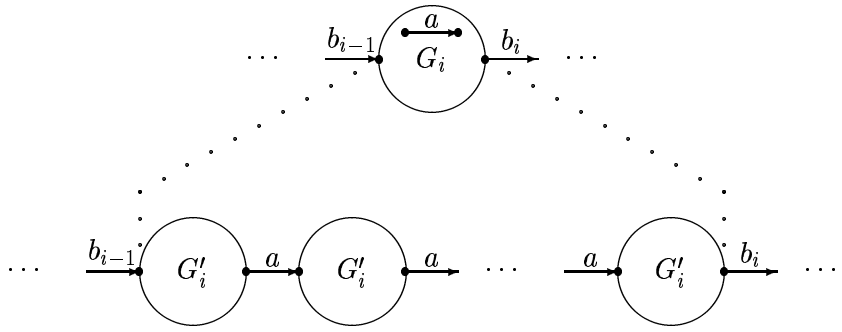


FIG. 2.3 – Réduction 1

réunion des promenades positives et contraintes des $C_{j,k}$. Si de plus j est non rigide dans G_i alors les tailles des $C_{j,k}$ sont inférieures à celle de C.

Pour $k \in \{1, \ldots, N\}$, on définit la chaîne $C_{j,k}$ comme une copie de C dans laquelle on remplace le SAVE $G_i = (Q_i, A_i)$, les arcs $b_{i-1} = (q_{i-1}, p_i)$ et $b_i = (q_i, p_{i+1})$ respectivement par :

i) le SAVE $\widetilde{G_i}$ définit comme suit :
   - l'ensemble des sommets est $Q_i \times \{0, \ldots, N\}$,
   - l'état d'entrée est $(p_i, (e_i)_j)$ et l'état de sortie est (q_i, k),
   - si $u = (r, s)$ est un arc de G_i alors $(u, l) = ((r, l), (s, l + (v(u))_j))$ est un arc de $\widetilde{G_i}$ de valuation celle de u sur toutes les coordonnées différentes de j et 0 sinon.

ii) auquel on ajoute l'arc $h_i = ((q_i, k), H)$ ayant pour valuation $k - (e_i)_j$ sur la j-ième coordonnée et nulle ailleurs, et le SAVE à un seul état H,

iii) l'arc $\widetilde{b_{i-1}} = (q_{i-1}, (p_i, (e_i)_j))$ de valuation celle de b_{i-1},

iv) l'arc $\widetilde{b_i} = (H, p_{i+1})$ de valuation celle de b_i,

On represente cette opération de réduction par le schéma suivant :

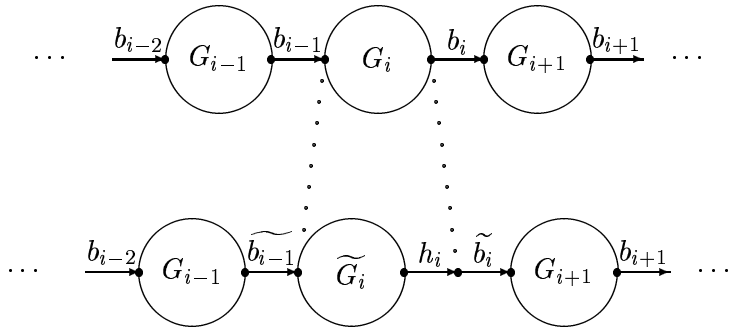


FIG. 2.4 – Réduction 2

La contrainte d'entrée de $\widetilde{G_i}$ est e_i, celle de H est $(\omega, \ldots, \omega)$ et la contrainte de sortie de $\widetilde{G_i}$ est $(\omega, \ldots, \omega)$, celle de H est s_i.

On dira que l'on a rigidifié C selon i sur la coordonnée j à l'ordre k.

2.4.5 Réduction de chaîne

Elle s'effectue à l'aide des opérations sur les chaînes. On introduit la définition d'un ensemble semi-linéaire jouant un rôle fondamental dans l'algorithme de Kosasraju.

Définitions 25 *Soit $G = (Q, A)$ un graphe tel que $A = \{a_i\}_{i\in\{1,\dots,n\}}$, on appelle* image commutative (ou trace) *d'un chemin c dans G le vecteur $\gamma(c) \in \mathbb{N}^m$ tel que la i-ème coordonnée a pour valeur le nombre d'occurence de a_i dans le chemin c.*
Soit C une chaîne ayant n SAVE et $P = (c_1 b_1 c_2 b_2 \dots b_{n-1} c_n, x)$ une promenade contrainte de C, on appelle trace étendue *de P le vecteur*

$$Tr^e(P) = (x_1, y_1, x_2, y_2, \dots, x_n, y_n, \gamma(c_1), \dots, \gamma(c_n)) \in \mathbb{N}^{m.2n.\Sigma_{i=1}^{n}|A_i|}$$

où x_i (resp. y_i) est le point d'entrée (resp. de sortie) de P dans G_i.

Théorème 19 *Soit C une chaîne, l'ensemble $L(C)$ des traces étendues des promenades contraintes de C est semi-linéaire.*

Théorème 20 *Si C est une chaîne ne vérifiant pas la propriété θ alors il existe un ensemble fini $\mathcal{R}$ de chaînes tel que :*
i) la taille de chaque $C' \in \mathcal{R}$ est strictement inférieure à celle de C,
ii) il existe une promenade positive et contrainte dans C si et seulement s'il existe une promenade positive et contrainte dans l'une des $C' \in \mathcal{R}$.

Voici la partie constructive de la preuve :

Preuve. Soit $\mathcal{R} = \emptyset$. Si $L(C) = \emptyset$ alors il n'y a pas de promenade contrainte et le théorème est vérifié, sinon

1. C ne vérifie pas la propriété $\theta 1$.
   Soit $L(C) = \cup_{f\in F} L_f$ l'ensemble des traces étendues des promenades contraintes de C où $L_f = L(b^f; d_1^f, \dots, d_\alpha^f)$. Soit $f \in F$, par hypothèse $\mathcal{J}(f) = \{j \mid (d_1^f + \dots + d_\alpha^f)_j = 0\}$ n'est pas vide. Soit $j \in \mathcal{J}(f)$ et $N = (b^f)_j$, on a donc j correspond soit à :
   - un arc a. On a alors que pour toute promenade contrainte de C de trace étendue dans L_f, le chemin associé n'utilise l'arc a qu'au plus N fois. On ajoute à $\mathcal{R}$ les chaînes $\{C_{j,k}^f\}_{k\in\{0,\dots,N\}}$ obtenues en déployant C selon l'arc a à l'ordre k,
   - une coordonnée non contrainte en entrée du SAVE G_i. On a alors que pour toute promenade contrainte de C de trace étendue dans L_f, le point d'entrée x_i dans G_i du chemin associé vérifie $(x_i)_j \leqslant N$. On ajoute à $\mathcal{R}$ les chaînes $\{C_{j,k}^f\}_{k\in\{0,\dots,N\}}$ copies de C dans lesquel on a remplacé la j-ème coordonnée de la contrainte d'entrée $(e_i)_j$ par k,

– une coordonnée non contrainte en sortie du SAVE G_i. De même que ci-dessus, on ajoute à $\mathcal{R}$ les chaînes $\{C^f_{j,k}\}_{k\in\{0,\dots,N\}}$ copies de C dans lesquel on a remplacé la j-ème coordonnée de la contrainte de sortie $(s_i)_j$ par k.

C est donc réduite en $\mathcal{R} = \bigcup_{f\in F}\bigcup_{j\in\mathcal{J}(f)}\bigcup_{k=0}^{N}\{C^f_{j,k}\}$.

2. C vérifie la propriété $\theta 1 \wedge \neg\theta 2$. Comme C vérifie $\theta 1$, chaque SAVE privé de ses points isolés est fortement connexe. On traite le cas où $\theta 2i)$ n'est pas vérifié, l'autre cas est symétrique (i.e. on fait de même sur la renversée C^{renv} obtenue en inversant le sens des arcs). D'après le théorème 13, il existe i un indice de SAVE tel que si on note $\mathcal{G}$ le graphe de couverture associé à l'arbre de Karp et Miller de G_i depuis la configuration $(p_i, (e_i)_{E_i\backslash R_i})$ et N le plus grand entier apparaissant dans $\mathcal{G}$, alors pour toute promenade contrainte de C, il existe $j \in E_i\backslash R_i$ tel que chaque point intermédiaire z dans G_i a sa j-ème coordonnée $(z)_j \leqslant N$.
Soit $B = (E_i\backslash R_i) \times \{0,\dots,N\}$, on définit $\mathcal{R}$ par les chaînes $\{C_{j,k}\}_{(j,k)\in B}$ obtenues en rigidifiant C selon i sur la coordonnée j à l'ordre k.

□

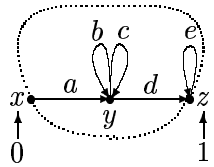
2.4.6 Algorithme de Kosaraju

Soient les vecteurs $x, y \in \mathbb{N}^m$ et C une chaîne ayant pour contrainte d'entrée x et de sortie y.

TANT QUE C ne satisfait pas la propriété θ ALORS
  SI il existe un élément dans $taille(C) \neq (0,0,0)$ ALORS
    on réduit C
  SINON (q_n, y) n'est pas $\mathbb{N}$–$accessible$ depuis (p_1, x)
    et on arrête l'algorithme.
(q_n, y) est $\mathbb{N}$–$accessible$ depuis (p_1, x).

Exemple

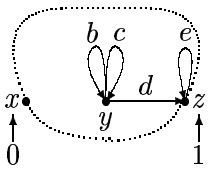
A-t-on $(x,0) \longrightarrow^+ (z,1)$ dans la chaîne ci-dessous avec $v(a) = v(b) = v(e) = 1$ et $v(c) = v(d) = -1$?



(La réponse est triviale en suivant par exemple le chemin ade.)
On a $taille(C) = ((1,5,0))$.

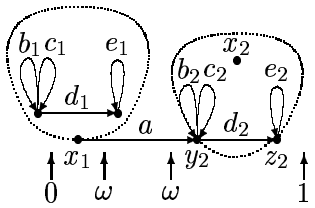
On voit qu'il n'existe pas de promenade contrainte utilisant 2 fois l'arc a. Donc C ne vérifie pas $\Theta 1$: on la réduit en déployant C selon l'arc a à l'ordre 0 et 1.

0 fois l'arc a :

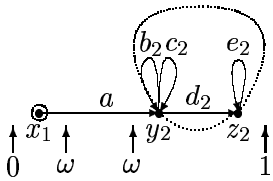


Or $L(C)$, l'ensemble des traces étendues de cette chaîne, est vide. En effet il n'y a pas de promenade contrainte. Donc on arrête l'algorithme dans cette branche.

1 fois l'arc a :

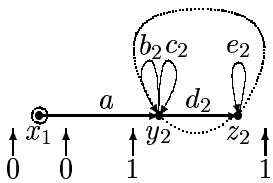


Or le calcul de $L(C)$ montre qu'aucuns arcs de G_1 n'est utilisé : on réduit en déployant la chaîne successivement selon chacuns de ces arcs. On obtient la chaîne suivante :



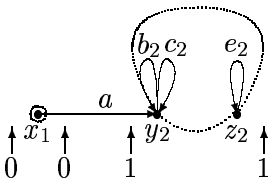
Il n'existe pas de promenade contrainte telle que la coordonnée non contrainte de sortie de G_1 est $\geqslant 0$: la propriété $\Theta 1ii)$ n'est pas vérifiée. Une coordonnée bornée de $L(C)$ correspond donc à cette coordonnée non contrainte (pour $N = 0$). On réduit en la même chaîne avec la contrainte d'entrée de G_1 égale à 0.

De même, il n'existe pas de promenade contrainte telle que la coordonnée non contrainte d'entrée de G_2 est $\geqslant 1$: on réduit en deux chaînes ayant la contrainte d'entrée de G_2 égale respectivement à 0 et à 1. Dans la première, il n'existe pas de promenade contrainte car on ne peut satisfaire les contraintes : on arrête l'algorithme dans cette branche. La seconde chaîne est la suivante :



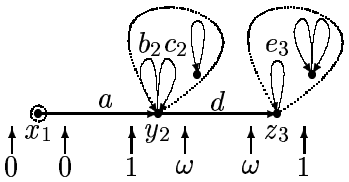
Comme il n'existe pas de promenade contrainte utilisant 2 fois l'arc d_2, on réduit la chaîne en déployant C selon l'arc d_2 à l'ordre 0 et 1.

0 fois l'arc d_2 :

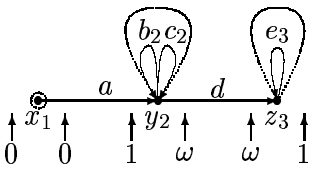


Or $L(C) = \emptyset$ donc on arrête l'algorithme dans cette branche.

1 fois l'arc d_2 :

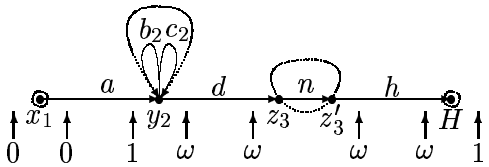


qui est réduite comme précédement en la chaîne :

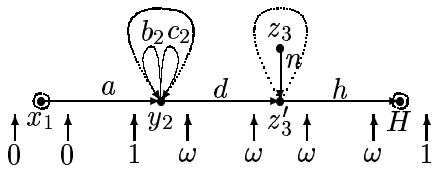


Cette chaîne vérifie $\Theta 1$ mais pas $\Theta 2$: $\forall \delta \geqslant 1$, $(z_3, 1)$ n'est pas $\mathbb{N}$-accessible depuis $(z_3, 1+\delta)$ dans le SAVE G_3'''. D'après le théorème thmK4.8, il existe $N = 1$ tel que tous les points intermédiaires de G_3''' sont majorés par 1. En effet le graphe de couverture depuis $(z_3, 1)$ dans $(G_3''')^{renv}$ a pour seul autre sommet la configuration $(z_3, 0)$ (voir la construction de la preuve du théorème 20).
Donc le réduit dépend de la valeur du point d'entrée en G_3''' :

si c'est zéro :

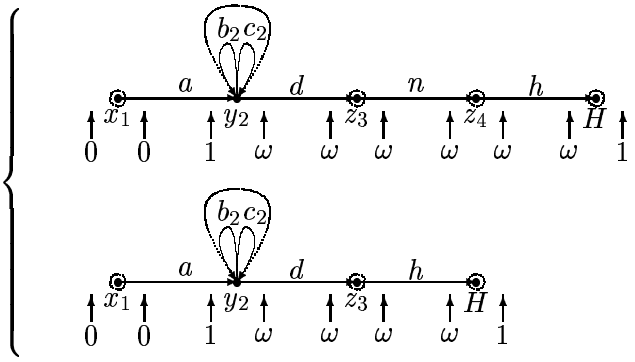


si c'est 1 :



avec $v(n) = 0$ et $v(h) = 1$.

On réduit ces chaînes ne vérifiant pas $\Theta 1$:



Elles vérifient alors $\Theta 1$ et comme toutes les coordonnées des derniers SAVEs sont rigides, elles vérifient aussi $\Theta 2$.

2.5 Preuve du théorème 9

Elle est obtenue par induction suivant l'arbre de recherche engendré par l'algorithme de Kosaraju, en appliquant les lemmes 22 et 24 décrits ci-après.

2.5.1 Chaîne vérifiant $\neg\theta 1$

Soit C une chaîne ne vérifiant pas $\theta 1$.
Soit $L(C) = \cup_{f\in F} L_f$ l'ensemble des traces étendues des promenades contraintes de C où $L_f = L(b^f; d_1^f, \ldots, d_\alpha^f)$.
Soit $\rho : (p_1, x_1) \longrightarrow (q_s, y_s)$ une promenade contrainte de C.
Soit $\mathcal{F}(\rho) = \{f \in F \mid Tr^e(\rho) \in L_f\}$.
Pour chaque $f \in F$, on définit $\mathcal{J}(f) = \{j \mid (d_1^f + \cdots + d_\alpha^f)_j = 0\}$.
Soit $\mathcal{B}(\rho) = \cup_{f\in\mathcal{F}(\rho)}\{(f, j, (b^f)_j) \mid j \in \mathcal{J}(f)\}$.

Lemme 21 *Soit C une chaîne ne vérifiant pas $\theta 1$ réduite en $\{C_{j,k}^f\}_{(f,j,k)\in B}$. Si ρ est une promenade contrainte de C alors pour tout $(f, j, k) \in \mathcal{B}(\rho)$, ρ est une promenade contrainte de $C_{j,k}^f$.*

Preuve. Si ρ est une promenade contrainte de C alors $L(C) = \cup_{f\in F} L_f = \{Tr^e(\rho) \mid \rho$ une promenade contrainte de $C\ \} \neq \emptyset$. Donc $\mathcal{F}(\rho) \neq \emptyset$ et on a $\forall f \in \mathcal{F}$, $\mathcal{J}(f) \neq \emptyset$ car C ne vérifie pas $\theta 1$. Soit $f \in \mathcal{F}$, on a pour tout $j \in \mathcal{J}(f)$ $(Tr^e(\rho))_j = (b^f)_j$ donc par construction des chaînes réduites de C, ρ est une promenade contrainte de $C_{j,k}^f$. □

Définition 26 *Pour tout triplet $(f, j, k) \in \mathcal{B}(\rho)$, le* relevé de la promenade contrainte $P = (\rho, x)$ *de C est la même promenade contrainte de $C_{j,k}^f$.*

La notion de relevé fait ici juste référence au fait que l'on regade une promenade non plus dans C mais dans certains de ces réduits (cf. lemme 21). On remarquera qu'une configuration d'une promenade contrainte de C est inchangée par relevé.

Lemme 22 *Soit C une chaîne ne vérifiant pas $\theta 1$.*
Si $\{\mathcal{A}^f_{j,k}\}_{(f,j,k)\in B}$ désigne les critères des chaînes réduites de C alors

$$\mathcal{A} = \left\{ (r,z) \;\middle|\; \begin{array}{l} \exists \rho_1 : (p_1,x_1) \longrightarrow (r,z)\ \forall \rho_2 : (r,z) \longrightarrow (q_s,y_s) \\ \forall (f,j,k) \in \mathcal{B}(\rho_1\rho_2)\ \textit{relevé}(r,z) \in \mathcal{A}^f_{j,k} \end{array} \right\}$$

est un critère de C.

Preuve. Soit C une chaîne ne vérifiant pas $\theta 1$ réduite en $\{C^f_{j,k}\}_{(f,j,k)\in B}$.
Soient $\{\mathcal{A}^f_{j,k}\}_{(f,j,k)\in B}$ leurs critères respectifs et $\mathcal{A}$ l'ensemble définit ci-dessus.
Soit $(r,z) \in \mathcal{A}$ et soit a un arc allant de r à r' dans C tel que $z + v(a) \geqslant 0$.
Par définition il existe $\rho_1 : (p_1,x_1) \longrightarrow (r,z)$ tel que pour tout $\rho_2 : (r,z) \longrightarrow (q_s,y_s)$, on a :

$$\forall (f,j,k) \in \mathcal{B}(\rho_1\rho_2)\ \text{relevé}(r,z) \in \mathcal{A}^f_{j,k}$$

Soit $(f,j,k) \in \mathcal{B}(\rho_1\rho_2)$, par définition on a relevé$(r,z) = (r,z)$.
or $(r,z) \in \mathcal{A}^f_{j,k}$ critère de $C^f_{j,k}$ et $a = (r,r')$ tel que $z' = z + v(a) \geqslant 0$ impliquent $(r',z') \in \mathcal{A}^f_{j,k}$ i.e. relevé$(r',z') \in \mathcal{A}^f_{j,k}$. Ainsi,

$$\begin{array}{l} \exists \rho_1 : (p_1,x_1) \longrightarrow (r,z)\ \forall a\rho_2' : (r,z) \xrightarrow{a} (r',z') \longrightarrow (q_s,y_s) \\ \forall (f,j,k) \in \mathcal{B}(\rho_1 a \rho_2')\ \text{relevé}(r',z') \in \mathcal{A}^f_{j,k} \end{array}$$

donc

$$\begin{array}{l} \exists \rho_1' = \rho_1 a : (p_1,x_1) \longrightarrow (r',z')\ \forall \rho_2' : (r',z') \longrightarrow (q_s,y_s) \\ \forall (f,j,k) \in \mathcal{B}(\rho_1'\rho_2')\ \text{relevé}(r',z') \in \mathcal{A}^f_{j,k} \end{array}$$

i.e. $(r',z') \in \mathcal{A}$. □

2.5.2 Chaîne vérifiant $\theta 1 \wedge \neg\theta 2$

Soit C une chaîne vérifiant $\theta 1 \wedge \neg\theta\ 2i)$ réduite suivant i en $\{C_{j,k}\}_{(j,k)\in B}$.
Soit $\mathcal{G}$ le graphe de couverture associé à l'arbre de Karp et Miller de G_i depuis la configuration $(p_i,(e_i)_{E_i\setminus R_i})$.
Soit $\rho : (p_1,x_1) \longrightarrow (q_s,y_s)$ une promenade contrainte de C.
Soit $J(\rho) = \{j \in (E_i\setminus R_i) \mid (z)_j < \infty\}$ où z est le point de la configuration obtenue dans $\mathcal{G}$ en suivant $\rho \downarrow_{G_i}$ depuis la racine.
Soit $\mathcal{B}(\rho) = \{(j,(y_i)_j) \in B \mid j \in J(\rho)\}$ où y_i est le point de sortie de ρ dans G_i.

Définitions 27 *Pour tout couple* $(j,k) \in K_{\mathcal{G}}(\rho)$, *le* relevé de la promenade contrainte $P = (\rho, x)$ de C *est la promenade contrainte* $\widetilde{P} = (\widetilde{\rho}, x)$ *de la chaîne* $C_{j,k}$ *telle que*

si on note $\rho = c_1 b_1 c_2 b_2 \dots b_{i-1} c_i b_i \dots c_s$ *alors*

$$\widetilde{\rho} = c_1 b_1 \dots c_{i-1} \widetilde{b_{i-1}} \widetilde{c_i} \widetilde{b_i} c_{i+1} \dots c_s$$

si on note $c_i = u_1 u_2 \dots u_t$ *alors* $\widetilde{c_i} = (u_1, (x_1)_j).(u_2, (x_2)_j). \dots .(u_t, (x_t)_j).h_i$

$$\text{où } \begin{cases} x_1 = x_i \text{ le point d'entrée de } \rho \text{ dans } G_i, \\ x_{a+1} = x_a + v(u_{a+1}) \text{ pour tout } a \in \{2, \dots, t-1\}, \end{cases}$$

et où $\widetilde{b_{i-1}}$,h_i *et* $\widetilde{b_i}$ *sont définis par la réduction (cf. 2.4.4).*
Pour tout couple $(j,k) \in \mathcal{B}(\rho)$, *le* relevé d'une configuration (r,z) *de la promenade* $P = (\rho, x)$ *contrainte de* C *est la configuration du relevé de* ρ *obtenue en suivant les arcs correspondants. On le note relevé*(r,z).

Si C est une chaîne vérifiant $\theta 1 \wedge \neg\theta 2 ii)$ alors on procéde de même avec sa renversée.

Lemme 23 *Soit* C *une chaîne vérifiant* $\theta 1 \wedge \neg\theta 2$ *se réduisant en* $\{C_{j,k}\}_{(j,k)\in B}$. *Si* ρ *est une promenade contrainte de* C *alors pour tout* $(j,k) \in \mathcal{B}(\rho)$, $\widetilde{\rho}$ *est une promenade contrainte de* $C_{j,k}$.

Preuve. Soit $\rho = c_1 b_1 c_2 \dots b_{i-1} c_i b_i \dots c_s, x_1)$ une promenade contrainte de C.
Soit $(j,k) \in B$, on a $k = (y_i)_j$ où y_i est le point de sortie de ρ dans G_i.

Par construction des $\{C_{j,k}\}_{(j,k)\in B}$, on a pour tout $i' < i$ $C \restriction_{G_{i'}} = C_{j,k} \restriction_{G_{i'}}$ et $\widetilde{\rho} \restriction_{G_{i'}} = \rho \restriction_{G_{i'}}$ donc la sous-promenade initiale de $\widetilde{\rho}$ allant de (p_1, x_1) à (q_s, y_s) est une promenade contrainte de la chaîne allant de G_1 à G_{i-1} dans $C_{j,k}$.

Or $v(\widetilde{b_{i-1}}) = v(b_{i-1})$ d'où le point d'entré de $\widetilde{\rho}$ dans $\widetilde{G_i}$ a même valeur que le point d'entré de ρ dans G_i, c'est à dire x_i.
- Relativement à la j-ème coordonnée :
  les arcs de $\widetilde{G_i}$ sont rigides i.e. leur valuation est nulle. Donc le point de sortie de $\widetilde{\rho}$ dans $\widetilde{G_i}$ a même valeur que le point d'entrée dans $\widetilde{G_i}$, c'est à dire $(x_i)_j$.
  Or h_i a pour valuation $k - (e_i)_j$ et $k = (y_i)_j$,
  or $(j,k) \in B$ assure que $j \in E_i \backslash R_i$, d'où $(x_i)_j = (e_i)_j$,
  donc $k - (e_i)_j = (y_i)_j - (x_i)_j$,
  donc le point d'entrée de $\widetilde{\rho}$ dans H a pour valeur $(x_i)_j + v(h_i)_j = (y_i)_j$,
  donc le point de sortie de $\widetilde{\rho}$ dans H a pour valeur sur la j-ème coordonnée $(y_i)_j$.

– Relativement aux autres coordonnées :
par construction, $\widetilde{\rho} \restriction_{\widetilde{G_i}}$ est une suite d'arcs correspondants à $\rho \restriction_{G_{i'}}$ et ayant même valuation. Donc le point de sortie de $\widetilde{\rho}$ dans $\widetilde{G_i}$ a même valeur que le point de sortie de ρ dans G_i.
Or h_i a une valuation nulle donc le point de sortie de $\widetilde{\rho}$ dans H qui est le point d'entrée de $\widetilde{\rho}$ dans H a pour valeur $(y_i)_{j'}$, pour tout $j' \neq j$.

Donc le point de sortie de $\widetilde{\rho}$ dans H a pour valeur y_i qui est le point de sortie de ρ dans G_i. Donc d'après les contraintes sur $\widetilde{G_i}$ et H, on a $\widetilde{\rho} \restriction_{\widetilde{G_i}\widetilde{h_i}H}$ est une promenade contrainte de $\widetilde{G_i}\widetilde{h_i}H$.

or $\widetilde{b_i}$ a pour valuation celle de b_i donc le point d'entrée de $\widetilde{\rho}$ dans G_{i+1} est $y_i + v(\widetilde{b_i}) = x_{i+1}$,
or $\forall i' > i$ on a $C \restriction_{G_{i'}} = C_{j,k} \restriction_{G_{i'}}$ et $\widetilde{\rho} \restriction_{G_{i'}} = \rho \restriction_{G_{i'}}$,
donc la sous-promenade finale de $\widetilde{\rho}$ allant de (p_{i+1}, x_{i+1}) à (q_s, y_s) est une promenade contrainte de la chaîne allant de G_{i+1} à G_s dans $C_{j,k}$.

Donc $\widetilde{\rho}$ est une promenade contrainte de $C_{j,k}$. □

Remarque Pour tout $(j,k) \in \mathcal{B}(\rho)$, ρ et $\widetilde{\rho}$ ne différent que sur G_i.

Lemme 24 *Soit C une chaîne vérifiant $\theta 1 \wedge \neg\theta 2$.*
Si $\{\mathcal{A}_{j,k}\}_{(j,k)\in B}$ désigne les critères des chaînes réduites de C alors

$$\mathcal{A} = \left\{ (r,z) \;\middle|\; \begin{array}{l} \exists \rho_1 : (p_1, x_1) \longrightarrow (r,z)\ \forall \rho_2 : (r,z) \longrightarrow (q_s, y_s) \\ \forall (j,k) \in \mathcal{B}(\rho_1\rho_2)\ \textit{relevé}(r,z) \in \mathcal{A}_{j,k} \end{array} \right\}$$

est un critère de C.

Preuve. Soit C une chaîne vérifiant $\theta 1 \wedge \neg\theta 2$ de critère $\mathcal{A}$ réduite selon i en $\{C_{j,k}\}_{(j,k)\in B}$ et $\{\mathcal{A}_{j,k}\}_{(j,k)\in B}$ leurs critères respectifs.
Soit $(r,z) \in \mathcal{A}$ et soit a un arc allant de r à r' dans C tel que $z + v(a) \geqslant 0$.
Par définition il existe $\rho_1 : (p_1, x_1) \longrightarrow (r,z)$ tel que pour tout $\rho_2 : (r,z) \longrightarrow (q_s, y_s)$, on a :

$$\forall (j,k) \in \mathcal{B}(\rho_1\rho_2)\ \text{relevé}(r,z) \in \mathcal{A}_{j,k}$$

Si $a \notin A_i \cup \{b_{i-1}, b_i\}$ alors d'après la remarque $\forall (j,k) \in \mathcal{B}(\rho_1\rho_2)$ relevé $(r,z) = (r,z)$,
or $(r,z) \in \mathcal{A}_{j,k}$ critère de $C_{j,k}$ et $a = (r,r')$ tel que $z' = z + v(a) \geqslant 0$ impliquent $(r',z') \in \mathcal{A}_{j,k}$,
or comme $a \notin A_i \cup \{b_{i-1}, b_i\}$, on a $(r',z') =$relevé(r',z'), $\forall (j,k) \in \mathcal{B}(\rho_1\rho_2)$.
Ainsi,

$$\begin{array}{l} \exists \rho_1 : (p_1, x_1) \longrightarrow (r,z)\ \forall a\rho_2' : (r,z) \xrightarrow{a} (r',z') \longrightarrow (q_s, y_s) \\ \forall (j,k) \in \mathcal{B}(\rho_1 a \rho_2')\ \text{relevé}(r',z') \in \mathcal{A}_{j,k} \end{array}$$

donc

$$\exists \rho'_1 = \rho_1 a : (p_1, x_1) \longrightarrow (r', z') \ \forall \rho'_2 : (r', z') \longrightarrow (q_s, y_s)$$
$$\forall (j,k) \in \mathcal{B}(\rho'_1 \rho'_2) \text{ relevé}(r', z') \in \mathcal{A}_{j,k}$$

i.e. $(r', z') \in \mathcal{A}$.

Si $a \in A_i \cup \{b_{i-1}, b_i\}$ alors on precéde de même modulo la notation utilisée dans $\widetilde{b_{i-1}} \widetilde{G_i} \widetilde{b_i}$. □

Bibliographie

[Asp87] A. Asperti. A logic for concurrency. Technical report, Dipartemento di informatica, Universitá di pisa, 1987.

[Bra83] G.W. Brams. *Réseaux de Pétri : thèorie et pratique*, volume 1 et 2. Masson, Paris, 1983.

[EN94] J. Esparza and M. Nielsen. Decidability issues for petri nets. Technical report, BRICS Report RS-94-8, 1994. ISSN 0909-0878.

[EW90] U. Engberg and G. Winskel. Petri nets as models of linear logic. In A. Arnold, editor, *Proceedings CAAP '90*, volume 431 of *Lecture Notes in Computer Science*, pages 147–161, Copenhagen, 1990. Springer-Verlag.

[GG89] C.A. Gunter and V. Gehlot. Nets as tensor theories. In G. De Michelis, editor, *Proc. 10-th International Conference on Application and Theory of Petri Nets, Bonn*, 1989.

[GS66] S. Ginsburg and E. H. Spanier. Semigroups, presburger formulas, and languages. *Pacific Journal of Mathematics*, 16(2) :285–296, 1966.

[GY80] A. Ginzburg and M. Yeoli. Vector addition systems and regular languages. *Journal of Computer and System Sciences*, 20 :227–284, 1980.

[Hau90] D. Hauschildt. Semilinearity of the reachability set is decidable for petri nets, 1990. FBI-HH-B-146/90.

[Kos82] S. R. Kosaraju. Decidability of reachability in vector addition systems. In *Proc. 14-th ACM Symp. on Theory of Computing*, pages 267–281, 1982.

[Lip76] R. Lipton. The reachability problem is exponential-space-hard. Technical Report 62, Dept. Comp. Science, Yale Univ., New Haven, CT, 1976.

[May81] E. W. Mayr. An algorithm for the general Petri net reachability problem. In *Proc. 13-th ACM Symposium on Theory of Computing, Milwaukee*, pages 238–246, 1981. Also in SIAM J. COMPUT., Vol.13(1994), No.3.

[May84] Mayr. An algorithm for the general petri net reachability problem. *Siam J. Computer*, 13 :441–460, 1984.

[Mog95] V. Mogbil. La décidabilité du problème d'accessibilité dans les réseaux de pétri, 1995. mémoire de D.E.A., Université d'Aix-Marseille II.

[MOM91] N. Martí-Oliet and J. Meseguer. From Petri nets to linear logic. *Mathematical Structures in Computer Science*, 1 :66–101, 1991. Revised version of paper in LNCS 389 (1989).

[Mül94] H. Müller. The reachability problem for vas. In *Advances in Petri Nets*, volume 188 of *Lecture Notes in Computer Science*, pages 376–391. Springer-Verlag, 1994.

[Reu89] Ch. Reutenauer. *Aspect mathématiques des réseaux de Petri*. Masson, Paris, 1989.

[VVN81] R. Valk and G. Vidal-Naquet. Petri nets and regular languages. *Journal of Computer and System Sciences*, 23 :299–325, 1981.

Chapitre 3

Codage des circuits hamiltoniens dans MLL

3.1 Introduction (français)

On a vu au chapitre 1.3 que le fragment multiplicatif de la logique linéaire est NP-Complet. En voici une preuve par le codage d'un problème de théorie des graphes : celui de décision des circuits hamiltoniens. C'est à dire, pour un graphe donné, existe-t-il un chemin passant exactement par tous les sommets (et qui revient à son origine) ? Considérons que l'on associe l'atome v_i au sommet V_i, le codage d'une arête (V_i, V_j) est donc une implication linéaire $(v_i \multimap v_j)$. Ainsi quand on construit le chemin hamiltonien cherché, on se pose la question de savoir quelle arête emprunter depuis le sommet courant. Il faut donc se donner la possibiliter d'"effacer" les arêtes que l'on ne choisit pas. La notion de choix est naturellement codée par les additifs : $(v_i \multimap v_j) \,\&\, 1$ code l'arête (V_i, V_j). Ce chapitre propose une alternative à la vision de choix en terme de connecteurs additifs, en codant ce probléme uniquement dans le fragment multiplicatif de la logique linéaire. Différentes preuves justifient ce codage dont une simple dans le fragment de Horn. Le codage étant présenté de façon intuitionniste, on a aussi la NP-Complétude de ce fragment. Pour conclure, l'intérêt réside d'une part dans la gestion multiplicative d'objets ayant un comportement typiquement additif. On peut envisager d'étendre ce genre d'intuition pour comprendre mieux la nature des connecteurs logiques. D'autre part ce résultat suggère d'autres études de problèmes de la théorie des graphes dans le contexte de la logique linéaire. Ce travail réalisé conjointement avec Thomas Krantz à donné lieu à une publication dans la revue "Theoretical Computer Science" sous le titre "Encoding hamiltonian circuits into multiplicative linear logic", dont le texte constitue la suite de ce chapitre.

3.2 Introduction

Max Kanovich proved the NP-completeness of various fragments of multiplicative linear logic (MLL) by an encoding of the 3-partition problem [Kan92]. We show the NP-completeness of MLL by encoding a problem of different nature, namely a graph-theoretical decision problem. This is a reference problem of the complexity theory. Our main contribution is to realize this without the use of additives. Normally a natural encoding of the hamiltonian circuit decision problem would be in the additive fragment (MALL), but this is not satisfactory because MALL is PSPACE-complete [LMSS92]. So we use a multiplicative management of the additives. We can find a similar idea in the proof of undecidability in the second order fragment of MLL [LS96] obtained from the result of Y.Lafont [Laf96] where the additives are used for zero-test. We give two proofs which justify our encoding, one using proof nets, and the other using Horn implications: we obtain an interpretation of the oriented graphs as formulas and of the paths as proofs. Since

the encoding is intuitionistic and MLL is conservative over its intuitionistic fragment, our result is also valid for intuitionistic multiplicative linear logic. Our approach suggests a more general study of (the foundations of) graph theory in the context of linear logic.

3.3 The encoding

Let G be an *oriented graph*. This means that G is a couple (V, E) with V being a finite non-empty set and $E \subseteq V \times V$. An element of V (respectively E) is called a *vertex* (respectively an *edge*). The first (respectively second) projection of an edge is called its *origin* (respectively *destination*). For a vertex i (respectively j) in V, we note $\deg^+(i)$ (respectively $\deg^-(j)$) the number of edges in G with origin i (respectively destination j).
A *path* from the vertex x to the vertex y in G is a sequence of edges $e_0, e_1, \ldots, e_l$ in G such that x is the origin of e_0, for every r, $0 \leq r < l$ the destination of e_r is the origin of e_{r+1} and y is the destination of e_l. A path p in G is *hamiltonian*, if every vertex of G occurs exactly once as the origin of an edge of p. A path from the vertex x to the vertex y is a *circuit* if $x = y$.
In the following we consider graphs[1] such that for each vertex i, $\deg^+(i) \geq 1$, $\deg^-(i) \geq 1$ and such that there is a vertex i, $\deg^+(i) \geq 2$ and a vertex j, $\deg^-(j) \geq 2$.
Let O be a vertex in V. Let V^* be the set $V - \{O\}$. To every vertex i in V we associate two atomic formulae a_i and b_i. It is easy to show that the existence of a hamiltonian circuit in G is equivalent to the provability in multiplicative additive linear logic of the sequent:

$$b_O, \{a_i \multimap b_i\}_{i \in V^*}, \{(b_i \multimap a_j) \,\&\, 1\}_{(i,j) \in E} \vdash a_O$$

Let k be an atomic formula, and $\mathcal{S}$ the sequent[2] of MLL

$$\{k \otimes a_i \multimap k \otimes b_i\}_{i \in V^*}, \{b_i \multimap a_j\}_{(i,j) \in E}, k \otimes a_O \multimap \otimes_{i \in V} b_i^{\delta_i^+} \vdash k \otimes b_O \multimap \otimes_{j \in V} a_j^{\delta_j^-}$$

where $\delta_i^+ = \deg^+(i) - 1$ for each vertex i and
$\delta_j^- = \deg^-(j) - 1$ for each vertex j.

Theorem 1 *There is a hamiltonian circuit in the oriented graph G if and only if the sequent $\mathcal{S}$ is provable in multiplicative linear logic.*

[1] The main result can be stated for graphs in general.

[2] $\otimes_{i \in V} x_i^{\delta_i}$ is equivalent to $\otimes_{i \in V'} x_i^{\delta_i}$ where V' is $\{i \in V | \delta_i \neq 0\}$. See for proof-nets with constants.

3.4 The condition is necessary

If G has a hamiltonian circuit $(v_0, v_1), (v_1, v_2), \ldots, (v_{n-1}, v_n)$, with $v_0 = v_n = O$ then we get by induction a proof of

$$\{k \otimes a_{v_i} \multimap k \otimes b_{v_i}\}_{i\in[1,\ldots,n-1]}, \{b_{v_i} \multimap a_{v_{i+1}}\}_{i\in[0,\ldots,n-1]}, k \otimes b_{v_0} \vdash k \otimes a_{v_n}.$$

If E^* is the set of edges not in the hamiltonian circuit, we get easily

$$\otimes_{i\in V} b_i^{\delta_i^+}, \{b_i \multimap a_j\}_{(i,j)\in E^*} \vdash \otimes_{j\in V} a_j^{\delta_j^-},$$

and we can finish the proof of $\mathcal{S}$ by a left and a right introduction of the linear implication.

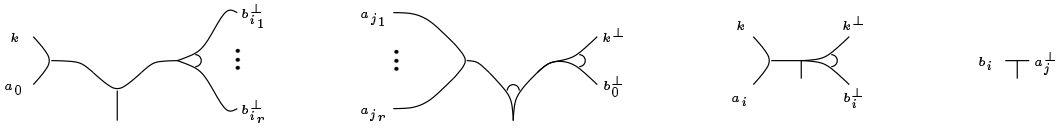
3.5 The condition is sufficient

3.5.1 Multiplicative proof-nets

We do not give full definitions for multiplicative proof-nets We use a modified version of the Danos-Regnier notation [DR89], and represent a binary tensor and par by: We use n-ary versions of the connectors as well. Remember

that the n-ary $\parr$ is considered as a single switch which is positioned on one of the premises. The Danos-Regnier correctness criterion for multiplicative proof-nets [DR89] is valid.

The following subnets, which correspond to the formulae[3] $k \otimes a_O \multimap \otimes_{i\in V} b_i^{\delta_i^+}$, $k \otimes b_O \multimap \otimes_{j\in V} a_j^{\delta_j^-}$, $k \otimes a_i \multimap k \otimes b_i$ and $b_i \multimap a_j$ are respectively called *F-device*, *I-device*, *V-device* and *E-device*[4].



3.5.2 Proof using proof-nets

Suppose given a proof-net $\mathcal{P}$ for the sequent $\mathcal{S}$. For a given atom A we will say that the device $\mathbf{d}_1$ is A-connected to the device $\mathbf{d}_2$ if there is an axiom-link connecting the A-port of $\mathbf{d}_1$ to the $A^\perp$-port of $\mathbf{d}_2$.

[3] The formulas on the left in the sequent $\mathcal{S}$ are negated.

[4] The notations stand respectively for final, initial, vertex and edge.

We use the correctness criterion to construct a hamiltonian circuit in G. Consider the k-axiom-links in $\mathcal{P}$. The acyclicity condition forbids the existence of a sequence of V-devices $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_l$, such that $\mathbf{v}_0 = \mathbf{v}_l$ and for each i, $0 \leq i < l$, $\mathbf{v}_i$ is k-connected to $\mathbf{v}_{i+1}$. It would be sufficient to put each $\parr$-switch occurring in one of the V-devices in the sequence on the position k to get a cycle. If we call $\mathbf{v}_0$ the I-device, then there is a sequence of V-devices $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$, such that for each i, $0 \leq i \leq n-1$, $\mathbf{v}_i$ is k-connected to $\mathbf{v}_{i+1}$, and $\mathbf{v}_n$ is the F-device. Every V-device $\mathbf{v}$ and the F-device is a_j-connected to an E-device.
If an E-device $\mathbf{e}$ is b_i-connected to the F-device, then the I-device is a_j-connected to $\mathbf{e}$, otherwise one would get a cycle by putting the $\parr$-switch of the F-device on the position corresponding to $\mathbf{e}$ and all other $\parr$-switches on V-devices on the k-position. From $\Sigma_{i \in V} \delta_i^+ = \Sigma_{i \in V} \delta_i^-$ we have that if the I-device is a_j-connected to an E-device $\mathbf{e}$, then $\mathbf{e}$ is b_i-connected to the F-device.
We prove by downward induction on the integer r, $r < n$ that if the V-device (or the F-device) $\mathbf{v}_{r+1}$ is a_j-connected to the E-device $\mathbf{e}_r$, and $\mathbf{e}_r$ is b_i-connected to a device $\mathbf{u}$, then $\mathbf{u}$ equals $\mathbf{v}_r$. If $\mathbf{u}$ is a $\mathbf{v}_l$, with $l < r$, by switching $\mathbf{v}_r$ on $a_j^\perp$, and $\mathbf{v}_l$ on $k^\perp$, we disconnect the proof-net. Thus $\mathbf{u}$ equals $\mathbf{v}_r$. The sequence $e_0, e_1, \ldots, e_{n-1}$, where e_l is the edge corresponding to the E-device $\mathbf{e}_l$, yields a hamiltonian circuit of G.

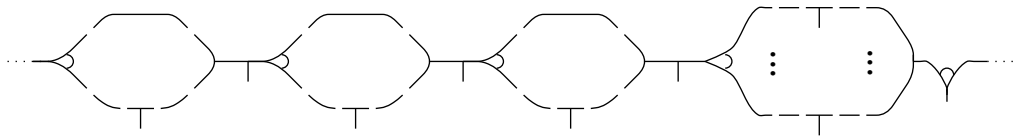


Figure 3.1: Example of a proof-net

3.5.3 Proof using horn programs

Definitions 1 *A* simple conjunction *is a tensor of positive literals.*
A Horn implication *is a formula of the form $X \multimap Y$ where X and Y are simple conjunctions.*

Definition 2 *For a multiset Γ consisting of Horn implications, a sequent of the form $W, \Gamma \vdash Z$ where W and Z are simple conjunctions is called a* Horn sequent.

Note that if $W = k \otimes b_O$, $Z = \otimes_{j \in V} a_j^{\delta_j^-}$ and

$$\Gamma = \left\{ \{(k \otimes a_i) \multimap (k \otimes b_i)\}_{i \in V^*}, \{b_i \multimap a_j\}_{(i,j) \in E}, (k \otimes a_O) \multimap \otimes_{i \in V} b_i^{\delta_i^+} \right\}$$

then $W, \Gamma \vdash Z$ is a Horn sequent. By reversibility of right linear implication, it is provable if and only if $\mathcal{S}$ is provable.
The idea of M.Kanovich [Kan92] is that a branching Horn program produces Z from W by consuming generalized Horn implications of Γ. Because our Γ is a multiset consisting only of Horn implications, we use a restricted form of Horn programs and suitable theorems.

Definition 3 *A* Horn program *is a chain where each vertex is labelled by a simple conjuction and each edge is labelled by a Horn implication $X \multimap Y$ which describes the elementary assignment operation producing $Y \otimes U$ from $X \otimes U$.*

Theorem 2 (Completeness[Kan92]) *For any Γ consisting of Horn implications, a sequent of the form*

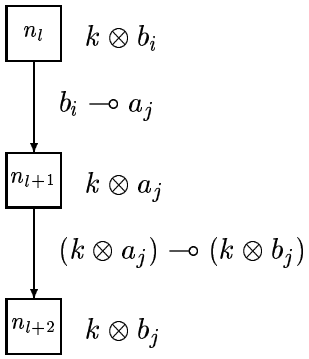
$$W, \Gamma \vdash Z$$

is derivable in linear logic if and only if we can construct a Horn program P such that

1. *All formulas used in the program P are from Γ,*
2. *In the chain P each formula of Γ is used exactly once,*
3. *The first node is labelled by W and the last one by Z.*

If the sequent $\mathcal{S}$ is provable then by the completeness theorem we can construct a Horn program P which satisfies:

- P starts from $k \otimes b_O$ and reaches $k \otimes a_O$ at a certain node using alternatively formulae of type $b_i \multimap a_j$ and $(k \otimes a_i) \multimap (k \otimes b_i)$:



- All the $\{(k \otimes a_i) \multimap (k \otimes b_i)\}_{i \in V^*}$ are used before one reaches $k \otimes a_O$.

Here is the key point of the proof: if a node in P has the label $k \otimes a_j$ then the next edge cannot have the labelling $(k \otimes a_O) \multimap \otimes_{i \in V} b_i^{\delta_i^+}$ before we have already used all of the $\{(k \otimes a_i) \multimap (k \otimes b_i)\}_{i \in V^*}$. Otherwise the next node has the label $\otimes_{i \in V} b_i^{\delta_i^+}$ which does not contain an occurence of k and then no following edge in the chain can be labelled by a $(k \otimes a_i) \multimap (k \otimes b_i)$. This contradicts the fact that in the chain P each formula of Γ is used exactly once. So this implies the existence of a hamiltonian circuit in $G = (V, E)$.

3.5.4 Proof using sequent calculus

In this section we work with proofs in intuitionistic sequent calculus.

Lemma 3 *Let $U \subseteq V$ and $E \subseteq V \times V$.*

i) If $k, b_i, \{k \otimes a_p \multimap k \otimes b_p\}_{p \in U}, \{b_p \multimap a_q\}_{(p,q) \in E} \vdash k \otimes a_j$ is provable with $\{i, j\} \not\subseteq U$ then there exists a hamiltonian path from i to j in $G = (U \cup \{i, j\}, E)$.

ii) If $k, a_i, \{k \otimes a_p \multimap k \otimes b_p\}_{p \in U}, \{b_p \multimap a_q\}_{(p,q) \in E} \vdash k \otimes a_j$ is provable with $i \in U$ and $j \notin U$ then there exists a hamiltonian path from i to j in $G = (U \cup \{j\}, E)$,

iii) If $k, b_i, \{k \otimes a_p \multimap k \otimes b_p\}_{p \in U}, \{b_p \multimap a_q\}_{(p,q) \in E} \vdash k \otimes b_j$ is provable with $i \notin U$ and $j \in U$ then there exists a hamiltonian path from i to j in $G = (U \cup \{i\}, E)$,

iv) If $k, a_i, \{k \otimes a_p \multimap k \otimes b_p\}_{p \in U}, \{b_p \multimap a_q\}_{(p,q) \in E} \vdash k \otimes b_j$ is provable with $\{i, j\} \in U$ then there exists a hamiltonian path from i to j in $G = (U, E)$.

Proof. By induction on $n = card(U) + card(E)$. Let $P(n)$ the conjunction of i) to iv) at rank n. Suppose that $P(m)$ is true for all $m < n$.
Case i): if $k, b_i, \{k \otimes a_p \multimap k \otimes b_p\}_{p \in U}, \{b_p \multimap a_q\}_{(p,q) \in E} \vdash k \otimes a_j$ is provable then consider the last rule in a cut-free proof of this sequent:

- rule of left linear implication on $k \otimes a_l \multimap k \otimes b_l$ for $l \in U$. Balance of atoms implies that the first sequent is provable if and only if

$$\begin{cases} k, b_i, \{k \otimes a_p \multimap k \otimes b_p\}_{p \in U_1}, \{b_p \multimap a_q\}_{(p,q) \in E_1} \vdash k \otimes a_l \\ k \otimes b_l, \{k \otimes a_p \multimap k \otimes b_p\}_{p \in U_2}, \{b_p \multimap a_q\}_{(p,q) \in E_2} \vdash k \otimes a_j \end{cases}$$

  are provable where $\{U_1, U_2\}$ is a partition of $U \backslash \{l\}$ and $\{E_1, E_2\}$ is a partition of E. By reversibility of left tensor rule and induction hypothesis i) there are hamiltonian paths from i to l in $G = (U_1 \cup \{i, l\}, E_1)$ and from l to j in $G = (U_2 \cup \{l, j\}, E_2)$. Because $\{i, j\} \not\subseteq U$, there exists a hamiltonian path from i to j in $G = (U_1 \cup U_2 \cup \{l, i, j\}, E_1 \cup E_2)$ i.e. in $G = (U \cup \{i, j\}, E)$.

- rule of left linear implication on $b_r \multimap a_s$ for $(r,s) \in E$. Balance of atoms implies that the first sequent is provable if and only if

$$\begin{cases} b_i, \{k \otimes a_p \multimap k \otimes b_p\}_{p\in U_1}, \{b_p \multimap a_q\}_{(p,q)\in E_1} \vdash b_r & (1) \\ k, a_s, \{k \otimes a_p \multimap k \otimes b_p\}_{p\in U_2}, \{b_p \multimap a_q\}_{(p,q)\in E_2} \vdash k \otimes a_j & (2) \end{cases}$$

  are provable where $\{U_1, U_2\}$ is a partition of U and $\{E_1, E_2\}$ is a partition of $E\backslash\{(r,s)\}$. By case analysis of the last rule, (1) is provable if and only if $i = r$ and $U_1 = E_1 = \emptyset$. By induction hypothesis ii) on (2), there is a hamiltonian path from s to j in $G = (U_2 \cup \{j\}, E_2)$. Because $i \notin U$, there exists a hamiltonian path from i to j in $G = (U_2 \cup \{i,j\}, E_2 \cup \{(r,s)\})$.

- rule of right tensor on $k \otimes a_j$. Balance of atoms implies that the first sequent is provable if and only if

$$\begin{cases} k, \{k \otimes a_p \multimap k \otimes b_p\}_{p\in U_1}, \{b_p \multimap a_q\}_{(p,q)\in E_1} \vdash k & (1) \\ b_i, \{k \otimes a_p \multimap k \otimes b_p\}_{p\in U_2}, \{b_p \multimap a_q\}_{(p,q)\in E_2} \vdash a_j & (2) \end{cases}$$

  are provable where $\{U_1, U_2\}$ is a partition of U and $\{E_1, E_2\}$ is a partition of E. It follows from a study of the last rule that (1) is provable if and only if $U_1 = E_1 = \emptyset$. Likewise (2) is provable if and only if $U_2 = \emptyset$ and $E_2 = \{(i,j)\}$ (*i.e.* $n = 1$). $G = (\{i,j\}, E_2)$ has a trivial hamiltonian path from i to j.

Case ii) is similar to case i) except that the last rule in a cut-free proof of this sequent can be a rule of right tensor on $k \otimes a_j$ if and only if $i = j$ and $U = E = \emptyset$ (*i.e.* $n = 0$). But also it cannot be a rule of left linear implication on $b_r \multimap a_s$ for $(r,s) \in E$ because atoms cannot be balanced.
Case iii) is similar to case i).
Case iv) is similar to case i) except that the last rule in a cut-free proof of this sequent cannot be a rule of left linear implication on $b_r \multimap a_s$ for $(r,s) \in E$ or a rule of right tensor on $k \otimes b_j$ because atoms cannot be balanced.
So $P(n)$ is true. □

Lemma 4 *Let $U \subseteq V$ and $E \subseteq V \times V$.*

i) If $k, b_i, \{k \otimes a_p \multimap k \otimes b_p\}_{p\in U^*}, \{b_p \multimap a_q\}_{(p,q)\in E}, (k \otimes a_O) \multimap \otimes_{i\in U} b_i^{\delta_i^+} \vdash \otimes_{j\in U} a_j^{\delta_j^-}$ is provable with $\{i\} \notin U$ then there exists a hamiltonian path from i to O in $G = (U \cup \{i, O\}, E)$,

ii) If $k, a_i, \{k \otimes a_p \multimap k \otimes b_p\}_{p\in U^*}, \{b_p \multimap a_q\}_{(p,q)\in E}, (k \otimes a_O) \multimap \otimes_{i\in U} b_i^{\delta_i^+} \vdash \otimes_{j\in U} a_j^{\delta_j^-}$ is provable with $i \in U$ then there exists a hamiltonian path from i to O in $G = (U \cup \{O\}, E)$.

Proof. By induction on $n = card(U) + card(E)$. Let $P(n)$ be i) and ii) at rank n. Suppose that $P(m)$ is true for all $m < n$.
Case i): if $k, b_i, \{k \otimes a_p \multimap k \otimes b_p\}_{p \in U^*}, \{b_p \multimap a_q\}_{(p,q)\in E}, (k \otimes a_O) \multimap \otimes_{i\in U} b_i^{\delta_i^+} \vdash \otimes_{j\in U} a_j^{\delta_j^-}$ is provable then consider the last rule in a cut-free proof of this sequent:

- rule of left linear implication on $k \otimes a_l \multimap k \otimes b_l$ for $l \in U^*$ and rule of left linear implication on $b_r \multimap a_s$ for $(r, s) \in E$. Similar to case i) of lemma 3, using reversibility of left tensor rule, induction hypothesis and lemma 3.

- rule of left linear implication on $(k \otimes a_O) \multimap \otimes_{i\in U} b_i^{\delta_i^+}$. Balance of atoms implies that the first sequent is provable if and only if

$$\begin{cases} \otimes_{i\in U} b_i^{\delta_i^+}, \{k \otimes a_p \multimap k \otimes b_p\}_{p\in U_1}, \{b_p \multimap a_q\}_{(p,q)\in E_1} \vdash \otimes_{j\in U} a_j^{\delta_j^-} & (1) \\ k, b_i, \{k \otimes a_p \multimap k \otimes b_p\}_{p\in U_2}, \{b_p \multimap a_q\}_{(p,q)\in E_2} \vdash k \otimes a_O & (2) \end{cases}$$

  are provable where $\{U_1, U_2\}$ is a partition of U^* and $\{E_1, E_2\}$ is a partition of E. By case analysis of the last rule, (1) is provable if and only if $U_1 = \emptyset$. Then $E_1 \subseteq U \times U$. By lemma 3 i) on (2) there is a hamiltonian path from i to O in $G = (U_2 \cup \{i, O\}, E_2)$. So there exists a hamiltonian path from i to O in $G = (U_2 \cup \{i, O\}, E_1 \cup E_2)$.

- rule of right tensor on $\otimes_{j\in U} a_j^{\delta_j^-}$ cannot appear by balance of atoms and considering possible rules. In fact a particular study is needed if the number of edges is two more than the number of vertices.

Case ii) is the same as case i) except that the last rule in a cut-free proof of this sequent cannot be a rule of left linear implication on $b_r \multimap a_s$ for $(r, s) \in E$ because atoms cannot be balanced.
So $P(n)$ is true. □

PROOF (encoding provable ⇒ existence of a hamiltonian circuit). By reversibility of the right linear implication and of the left tensor rule, provability of $\mathcal{S}$ implies that the hypothesis of lemma 4 i) with $U = V$ is satisfied. So there is a hamiltonian path from O to O in $G = (V^* \cup O, E)$ *i.e.* there exists a hamiltonian circuit in $G = (V, E)$. □

3.6 Conclusion

The encoding should give some intuition for a multiplicative management of additives in other cases as well. We remark that we use a small fragment of

multiplicative linear logic. In fact with a slight modification of the encoding the valid proof-nets are planar. The question of NP-completeness of non-commutative multiplicative linear logic[5] remains open though, as no order is imposed *a priori* on the formulae of the sequent $\mathcal{S}$.

Acknowledgements
We thank Yves Lafont, who made useful contribution to this work and the team "Logique de la Programmation" of the IML, Marseille, for its encouraging support and ideal conditions for doing research.

3.7 Appendix

3.7.1 Sequent calculus for intuitionistic multiplicative linear logic

A formula is either a positive atom A, or a negative one $A^{\perp}$, or a constant 1, or constructed using binary connectors $A \otimes B$ (tensor), $A \multimap B$ (linear implication). Intuitionistic sequents are of the form $\Gamma \vdash A$ where Γ is a multiset of formulae and A a formula. The rules for the intuitionistic sequent calculus are the following:

Identity group

$$\frac{}{A \vdash A}\ \text{(identity)} \qquad \frac{\Gamma \vdash A \quad A, \Delta \vdash B}{\Gamma, \Delta \vdash B}\ \text{(cut)}$$

Logic group

unit:

$$\frac{}{\vdash 1}\ \text{(one)}$$

tensor:

$$\frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}\ \text{(left)} \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\ \text{(right)}$$

linear implication:

$$\frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \multimap B \vdash C}\ \text{(left)} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\ \text{(right)}$$

3.7.2 Some properties

- Classical multiplicative linear logic is conservative over intuitionistic multiplicative linear logic (see [GLR95] for definitions): an intuitionistic sequent is provable in the intuitionistic calculus if and only if it is classically provable.
- The calculus verifies cut elimination, so a provable sequent has a proof not using the cut rule.

[5] see [Pen97] for a partial result.

- A rule is *reversible* if the provability of its conclusion implies the provability of its premises. The left tensor rule and the right linear implication rule are reversible.

- Balance of atoms: if we define $p_A(A) = 1, p_A(B \otimes C) = p_A(B) + p_A(C)$ and $p_A(B \multimap C) = p_A(C) - p_A(B)$ for an atom A then every provable sequent $B_1, \ldots, B_n \vdash C$ satisfies $p_A(B_1) + \ldots + p_A(B_n) = p_A(C)$.

Bibliographie

[BM79] J.A. Bondy and U.S.R. Murty. *Graph theory and related topics.* Academic Press, 1979.

[DR89] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28 :181–203, 1989.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability : a guide to the theory of NP-completeness.* Freeman,W.H. and Company, San Francisco, 1979.

[GLR95] J.-Y. Girard, Y. Lafont, and L. Regnier, editors. *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Notes*. Cambridge University Press, 1995. Proceedings of the 1993 Workshop on Linear Logic, Cornell University, Ithaca.

[Kan92] Max I. Kanovich. Horn programming in linear logic is NP-complete. In *Proceedings 7th Annual IEEE Symposium on Logic in Computer Science, LICS'92, Santa Cruz, CA, 22–25 June 1992*, pages 200–210. IEEE Computer Society Press, Los Alamitos, California, 1992.

[Laf96] Y. Lafont. The undecidability of second order linear logic without exponentials. *Journal of Symbolic Logic*, 61 :541–548, 1996.

[LMSS92] P. Lincoln, J. Mitchell, A. Scedrov, and N. Shankar. Decision problems for propositional linear logic. *Annals of Pure and Applied Logic*, 56 :239–311, 1992. Also in the Proceedings of the 31th Annual Symposium on Foundations of Computer Science, St Louis, Missouri, October 1990, IEEE Computer Society Press. Also available as Technical Report SRI-CSL-90-08 from SRI International, Computer Science Laboratory.

[LS96] Y. Lafont and A. Scedrov. The undecidability of second order multiplicative linear logic. *Information and Computation*, 125(1) :46–51, 1996.

[Pen97] M. Pentus. Equivalence of multiplicative fragments of cyclic linear logic and noncommutative linear logic. In S. Adian and al., editors, *LFCS'97 (Yaroslavl, Russia, 1997)*, volume 1234 of *Lecture Notes in Computer Science*, pages 306–311, 1997.

Chapitre 4

Critère de correction quadratique pour NL

4.1 Introduction (français)

On a introduit au chapitre 1.2.3 la notion de réseaux de preuve de la logique linéaire que l'on peut séquentialiser en preuves du calcul des séquents. Ces réseaux sont des structures de preuves (construites sur des graphes) reconnues par un critère de correction. Initialement J.-Y. Girard s'est servit de critère utilisant les longs voyages [Gir87], mais le plus utilisé est certainement celui dit de Danos-Regnier [DR89]. S. Guerrini a montré que la correction des réseaux de preuve multiplicatifs se fait en temps linéaire [Gue99]. On est loin de ce résultat en logique non-commutative multiplicative (MNL) : on connait un critère utilisant les longs voyages [AR00] et récemment R. Maieli en a montré un du type de celui de Danos-Regnier [Mai00]. Cependant ils sont en temps exponentiel.

Si l'on regarde le résultat de S. Guerrini pour MLL, il fait intervenir une reformulation du critère de contraction de V. Danos [Dan90] d'abord sous forme d'algorithme de "parsing" puis en terme d'unification. Son implémentation donne un algorithme quasi-linéaire[1]. La linéarité du critère est obtenu par une étude algorithmique fine. Le travail présenté ici est une première étape dans cette direction. On établit des critères de contraction pour MNL qui présentent l'avantage d'être en temps quadratique. Ils se présentent sous la forme de régles de réécriture inspirées par le calcul des séquents de MNL. On établit des résultats de confluence et de séquentialisation.

Pour avoir une intuition sur le fonctionnement du premier de ces critères, il faut revenir à la structure de variété d'ordre série-parallèle permettant de décrire les séquents de NL. Elle procure un point de vue (i.e. on se positionne depuis un élément x de l'ensemble de base) qui peut être regardé comme un ordre partiel sur le complémentaire de $\{x\}$. Les variétés d'ordre peuvent être présentées de différentes façons en changeant de point de vue mais sont invariantes par un tel changement (l'analalogie classique étant qu'un cercle orienté devient un ordre total dès qu'une origine est fixée). Cette idée correspond à la capacité d'isoler une formule de son contexte pour lui appliquer une régle. Elle permet aussi une forme de cyclicité et la restriction du calcul des séquents à la classe des variétés d'ordre totale est exactement CyLL. Comme on le verra un peu plus loin, les variétés d'ordre peuvent être représentées à l'aide d'arbre sans racine : les algues. Les noeuds sont de type commutatif ou non-commutatif. L'associativité de ces noeuds permet une représentation sous forme normale : les types de noeuds sont alternés. L'un des critère consiste donc naturellement à contracter une structure de preuve de MNL en une algue en se servant de régles de réécriture, basées sur celles du calcul des séquents, appliquées non pas sur des variétés d'ordre mais sur leur représentation sous forme d'algues. On définit ainsi des structures de

[1] En fait α-linéaire où α est l'inverse de la fonction d'Ackerman, c'est à dire une fonction trés faiblement croissante. Dans toute implémentation concevable $\alpha(h,k) < 4$

preuves contractibles mélangeant structures de preuves (construites sur les liens de MNL) et algues. Le lien axiome d'une structure de preuve étant une arrête dans l'algue, le lien tenseur $\otimes$ un noeud commutatif et le lien next $\odot$ (conjonction non-commutative) un noeud non-commutatif. Pour espérer avoir un système confluent il faut garder l'idée présente dans les réseaux de MNL : ne pas avoir d'entropie explicite. L'entropie permet par affaiblissement sur les variétés d'ordre de traiter commutativement quelque chose qui ne l'est pas. Cependant les seules régles qui peuvent nécéssiter préalablement l'usage d'entropie sont celles du tenseur $\otimes$ et du par $\parr$ (i.e. les régles commutatives). Le cas du tenseur étant trivialisé par le choix de la structure d'algue, il ne reste que la régle du par. On montre qu'entre deux formules étiquettant une algue, l'application d'entropie de façon minimale correspond exactement à une opération de substitution sur les variétés d'ordre. La contraction d'un lien par d'une structure de preuve contractible est donc l'algue représentant la variété d'ordre substituée.
Le second critère de contraction utilise directement les variétés d'ordre plutôt que leur représentation sous forme d'algue. On contracte un réseau de preuve en un noeud généralisé étiquetté par la variété d'ordre correspondante. Ces régles de réécriture sont directement décrites comme un algorithme de "parsing" : elles constituent la deuxième étape vers une éventuelle amélioration de la compléxité de la correction des réseaux de preuve de MNL. Ces résultats ont faits l'objet d'une soumission au sixième symposium annuel de "Logic In Computer Science" (LICS 2001) sous le titre "quadratic correctness criteria for non-commutative logic" dont le texte en anglais constitue la première partie.
Dans seconde partie on étend le critère de contraction au calcul des séquent avec règle de coupure sans en changer la complexité. On montre que l'on a alors de bonnes propriétées comme la préservation de la correction par élimination des coupures (c'est un résultat non trivial dans le cas non commutatif) et qu'un réseau de preuve de MNL est acyclique et connexe. On retrouve bien sûr un théorème d'adéquation et de séquentialisation. Le lien est fait de nouveau avec le précédent algorithme que l'on étend de la même façon. Cela amène à une discussion sur la règle de coupure en liaison avec les problèmes rencontrés dans les critères de [AR00] et [Mai00]. Enfin la traduction commutative de ce critère de contraction donne le crière de contraction de V. Danos.

4.2 Introduction

Non commutative Logic (NL) is a unification of linear logic [Gir87] and cyclic linear logic [Gir89, Yet90, Abr91] (a classical conservative extension of the Lambek calculus [Lam58]). It includes all linear connectives : multiplicatives, additives, exponentials and constants. Recents results [AR00, Rue00, MR00] introduce proof nets, sequent calculus, phase semantics and all the importants theorems like cut elimination and sequentialisation. The central notion is the structure of order varieties. They can be presented in different ways by changing the point of view and are invariant under the change of presentation : one uses rootless planar trees called seaweeds. Thus this structure allows focusing on any formula to apply a rule.

Proof nets are graph representations of NL derivations. A proof net with conclusion A can be sequentialized : the corresponding cut-free derivation of the formula A is not unique in general. It introduces some irrelevant order on the sequent rules. For instance, a derivation Π ending with $\vdash A \parr B, C \parr D$ implies an order on the two rules introducing the principal connectives of $A \parr B$ and $C \parr D$, but the proof net corresponding to Π does not depend on such order.

A contracting proof structure is a hypergraph built in accordance with the syntax of proof nets and seaweeds. A proof structure is a particular contracting one. To know if a such structure is a proof net or not, we use a correctness criterion. The Maieli one is in the Danos-Regnier criterion style : at first it uses a switching condition and tests if we obtain an acyclic connected graph. Then for each ∇ link, we check the associated order varieties.

It is known that the proof nets of multiplicative linear logic have a linear time correctness criterion [Gue99]. The first step towards a linear algorithm is to have a contractibility criterion (the Danos one [Dan90]) which can be seen as a parsing algorithm. One can reformulate it in terms of a sort of unification. Then a direct implementation leads a quasi-linear algorithm, and sharp study give the exact complexity. Up to now, there was no polymomial criterion for MNL.

Here we present a set of shrinking rules for MNL proof structures characterising MNL proof nets as the only structures that contract to a seaweed. We show that this contractibility criterion is quadratic. This idea is extended by a presentation as a parsing algorithm. So this work may be a decisive step towards a linear MNL correctness criterion.

Notations : One writes $X \uplus Y$ for the disjoint union of the sets X and Y. Let ω and τ orders respectively on the sets X and Y. Let $x \in X$. One writes $\omega[\tau/x]$ the order on $(X \backslash \{x\}) \cup Y$ defined by $\omega[\tau/x](y,z)$ iff $\omega(y,z)$ or $\tau(y,z)$ or $\omega(y,x)$ if $z \in Y$ or $\omega(x,z)$ if $y \in Y$. Let f and g be positive functions. One writes $g(n) = \Theta(f(n))$ to denote that $f = \mathcal{O}(g)$ and $g = \mathcal{O}(f)$.

4.3 Order varieties

4.3.1 Order varieties and orders

Definition 4 (order varieties) *Let X be a set. An* order variety *on X is a ternary relation α which is :*

$$\begin{cases} \textit{cyclic :} & \forall x,y,z \in X, \alpha(x,y,z) \Rightarrow \alpha(y,z,x), \\ \textit{anti-reflexive :} & \forall x,y \in X, \neg\alpha(x,x,y), \\ \textit{transitive :} & \forall x,y,z,t \in X, \alpha(x,y,z) \textit{ and } \alpha(z,t,x) \Rightarrow \alpha(y,z,t), \\ \textit{spreading :} & \forall x,y,z,t \in X, \alpha(x,y,z) \Rightarrow \alpha(t,y,z) \textit{ or } \alpha(x,t,z) \textit{ or } \alpha(x,y,t). \end{cases}$$

Definition 5 (series-parallel orders) *Let ω and τ two partial orders on disjoints sets X and Y respectively. Their* serial sum *(resp.* parallel sum*) $\omega < \tau$ (resp. $\omega \parallel \tau$) is a partial order on $X \cup Y$ defined respectively by :*

$$\begin{array}{lll} (\omega < \tau)(x,y) & \textit{iff} & x <_\omega y \textit{ or } x <_\tau y \textit{ or } (x \in X \textit{ and } y \in Y), \\ (\omega \parallel \tau)(x,y) & \textit{iff} & x <_\omega y \textit{ or } x <_\tau y. \end{array}$$

Definition 6 (closure) *Let $\omega = (X, <)$ be a partial order on X and $z \in X$. Let $\overset{z}{<}$ denote the binary relation : $x \overset{z}{<} y$ iff $x < y$ and z is comparable neither with x nor y. The* closure *of ω is the ternary relation $\overline{\omega}$ on X defined by :*

$$\begin{array}{lllllllll} \overline{\omega}(x,y,z) & \textit{iff} & x<y<z & \textit{or} & y<z<x & \textit{or} & z<x<y & \textit{or} \\ & & x \overset{z}{<} y & \textit{or} & y \overset{x}{<} z & \textit{or} & z \overset{y}{<} x. & \end{array}$$

Facts 1 i) If ω is a partial order on X then $\overline{\omega}$ is an order variety on X,
ii) The closure identifies serial and parallel sums of partials orders on disjoints sets..

Definition 7 (gluing) *Let ω and τ two partial orders on disjoints sets X and Y respectively. The* gluing *$\omega * \tau$ of ω and τ is the following order variety on $X \cup Y$:*

$$\omega * \tau = \overline{\omega < \tau} = \overline{\omega \parallel \tau} = \overline{\tau < \omega}$$

Definition 8 *Let α be an order variety on a set X and $x \in X$. The order* α_x induced by α and x *is the partial order on $X \backslash \{x\}$ defined by :*

$$\alpha_x(y,z) \textit{ iff } \alpha(x,y,z)$$

One writes x for the unique partial order on $\{x\}$.

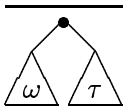
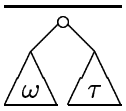
Proposition 5 *Let α be an order variety on a set X, $x \in X$ and ω a partial order on $X \backslash \{x\}$. Then*

$$\alpha_x * x = \alpha \quad \textit{and} \quad (\omega * x)_x = \omega$$

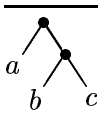
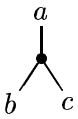
Fact 2 Let α be an order variety on a non-empty set. α is series-parallel iff there exists a series-parallel order ω such that $\alpha = \overline{\omega}$. In other words, series-parallel order varieties are exactly those can be represented by series-parallel orders.

Definition 9 (seaweed) *Let $\alpha = \overline{\omega}$ be a series-parallel order variety on X ($\sharp X \geq 2$) such that ω is written as a (non-unique) binary tree T with leaves labelled by elements of X, and root and nodes labelled by $\bullet$ (serial composition) or $\circ$ (parallel composition).*
A seaweed *S representing α is a rootless planar tree with leaves labeled by elements of X and ternary nodes labeled by $\bullet$ or $\circ$, defined by removing the root of T :*

$$\alpha \;=\; \overline{\omega < \tau} \;=\; \omega * \tau \;=\; \overline{\omega \parallel \tau}$$

$$=$$  $$=$$  $$=$$ 

By convention orders are represented with top root and then seaweeds are oriented anti-clockwise :

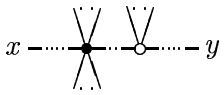
$$\overline{a < b < c} \;=$$  $$=$$ 

One extends the definition of seaweeds to the rootless planar trees on n-ary-nodes ($n \geq 3$).

Definition 10 (normal form) *Let α be a series-parallel order variety. Let the seaweeds representing α be considered modulo assocativity of $\circ$ and $\bullet$: there is not two nodes linked with a same label, and there is not binary or unary nodes. The equivalence class of such seaweeds modulo commutativity of $\circ$ has a unique representative which is said in* normal form.

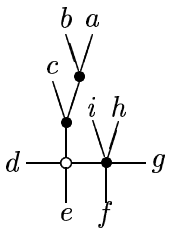
The uniqueness comes from the next proposition.

Remark. A seaweed is in normal form if it has n-ary nodes and verifies that all paths between two leaves are a sequence of alternate $\bullet$ and $\circ$ nodes. Afterwards for a seaweed (not specially in normal form) we denote such alternates paths between arbitrary leaves x and y by the following figure :



This notation does not presuppose that this alternate path starts by a •-node and finishes by a ∘-node.

Example: Let α be the closure of $[((a < b) < c) \parallel d] < [e \parallel (f < (g < h) < i)]$. Then the path between d and g is



To be convenient we only use seaweeds in normal form. So ∘-nodes are commutative. When it is not ambiguous, we use an order variety instead of its representation.

4.3.2 Seesaw and entropy

Definitions 11 *Let ω and τ be series-parallel orders on a same given set. The equivalence relation* seesaw *is defined by $\overline{\omega} = \overline{\tau}$. The relation* entropy $\preccurlyeq$ *is defined by $\omega \preccurlyeq \tau$ iff $\omega \subseteq \tau$ and $\overline{\omega} \subseteq \overline{\tau}$.*

Proposition 6 *In the case of series-parallel orders, seesaw (resp. entropy) turn out to be the least equivalence $\backsim$ (resp. the least reflexive transitive relation) given by :*

$$(\omega_1 \parallel \omega_2) \backsim (\omega_1 < \omega_2) \qquad \textit{(resp. } \omega[\omega_1 \parallel \omega_2] \preccurlyeq \omega[\omega_1 < \omega_2] \textit{)}$$

Facts 3 i) Entropy is a partial order, compatible with restriction and the serial and parallel sums of orders,

ii) entropy between orders corresponds to inclusion of order varieties : let α and β be order varieties on X, and $x \in X$, we have

$$\alpha \subseteq \beta \text{ iff } \alpha_x \preccurlyeq \beta_x.$$

This is independent from the choice of x,

iii) entropy is performed on seaweeds by changing some •-nodes into ∘-nodes.

4.3.3 Wedge and identification

Definitions 12 (wedge) *Let $(\omega_i)_{i \in I}$ be a non empty family of partials orders on a same set. The* wedge $\bigwedge_{i \in I} \omega_i$ *is a largest partial order (w.r.t. $\preccurlyeq$) such that*

$$(\bigwedge_{i \in I} \omega_i) \preccurlyeq \omega_i \textit{ for all } i \in I.$$

Let $(\alpha_i)_{i\in I}$ be a non empty family of order varieties on a set X. The wedge $\bigwedge_{i\in I}\alpha_i$ *is*

$$(\bigwedge_{i\in I}(\alpha_i)_x) * x$$

for an arbitrary $x \in X$.

Facts 4 i) Partials orders on a given set form a complete inf-semi-lattice for entropy and wedge,
ii) the wedge is not intersection in general,
iii) the wedge is not series-parallel in general, even if all ω_i are series-parallel,
iv) the wedge (partially) commutes with restriction :

$$\text{if } Y \subseteq |\omega_i| \text{ then } (\bigwedge_{i\in I}\omega_i) \restriction Y \trianglelefteq (\bigwedge_{i\in I}\omega_i \restriction Y),$$

v) the two notions of wedge are related by :

$$(\bigwedge_{i\in I}\alpha_i)_x = \bigwedge_{i\in I}(\alpha_i)_x \quad \text{and} \quad (\bigwedge_{i\in I}\omega_i) * x = \bigwedge_{i\in I}(\omega_i * x)$$

Definition 13 (identification) *Let α be an order variety on a set $X \uplus \{x\} \uplus \{y\}$, and let $z \notin X \cup \{x, y\}$. The* identification $\alpha[z/x, y]$ *of x and y into z in α is the order variety defined by :*

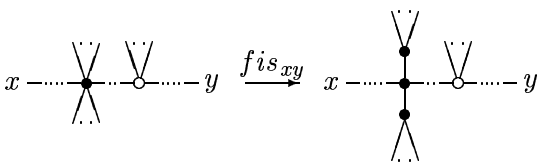
$$\alpha[z/x, y] = \alpha \restriction_{X\cup\{x\}} [z/x] \ \wedge \ \alpha \restriction_{X\cup\{y\}} [z/y]$$

Lemma 7 *i) $\alpha[z/x, y]_z * (x \parallel y) \subseteq \alpha$,*
ii) Let α be an order variety on $X \uplus \{x\} \uplus \{y\}$ and ω be a partial order on X such that $\omega * (x \parallel y) \subseteq \alpha$. Then $\omega * (x \parallel y) \subseteq \alpha[z/x, y]_z * (x \parallel y)$, or equivalently $\omega \trianglelefteq \alpha[z/x, y]_z$.

Proof. See the proof of lemma 3.35 in [Rue00]. □

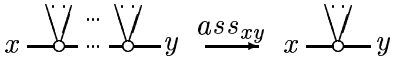
Definition 14 *Let α be a series-parallel order variety represented by a seaweed S. We define the seaweed $S\langle z/x, y\rangle$ by the following sequence on the alternate path between x and y in S :*

1. *fis_{xy} : transform every $\circ$-node belong the path between x and y. This is called "fission" :*



2. ent_{xy} : apply entropy belong the path between x and y :

3. ass_{xy} : apply associativity belong the path between x and y :



4. substitute z for $x \parallel y$.

Lemma 8 *i) Identification in order varieties is monotonic (for the inclusion),*
ii) If v denote a map such that $v(s)$ is the order variety corresponding to the seaweed S then, for S and T seaweeds,

$$v(S) \subseteq v(T) \implies v(S\langle z/x, y\rangle) \subseteq v(T\langle z/x, y\rangle)$$

Proof. Let α and β be order varieties on a set X such that $\alpha \subseteq \beta$. We have $\alpha[z/x, y] \subseteq \beta[z/x, y]$ i.e. identification is monotonic because the wedge is clearly monotonic. On the seaweeds, the only nodes which are different in the representation of α and β are the $\circ$-nodes in the representation of α which correspond to $\bullet$-nodes in the representation of β. If so,
- by definition, for all $x, y \in X$, $fis_{xy}(\alpha)$ and $fis_{xy}(\beta)$ represent always the same included order varieties,
- all differents nodes on the path between x and y become $\circ$-nodes in $ent_{xy}(fis_{xy}(\alpha))$ and stay $\circ$-nodes in $ent_{xy}(fis_{xy}(\beta))$,
- all others are unchanged.

Hence the order variety represented by $ent_{xy}(fis_{xy}(\alpha))$ is included in the one which is represented by $ent_{xy}(fis_{xy}(\beta))$ □

Proposition 9 *Let α be a series-parallel order variety on a set $X \uplus \{x\} \uplus \{y\}$, and let $z \notin X \cup \{x, y\}$. If the seaweed S represents α then the seaweed $S\langle z/x, y\rangle$ represents the identification $\alpha[z/x, y]$.*

Proof. Using the notations of lemma 8,
$\supseteq$) With the hypothesis, we have that $\alpha[z/x, y]_z * (x \parallel y) \subseteq \alpha$. Then by the previous lemma,

$$v((\alpha[z/x, y]_z * (x \parallel y))\langle z/x, y\rangle) \subseteq v(\alpha\langle z/x, y\rangle)$$

So by definition of $S\langle z/x, y\rangle$, we obtain that

$$\alpha[z/x, y]_z * z \subseteq v(\alpha\langle z/x, y\rangle)$$

For all $u \in |\alpha|$ $\alpha_u * u = \alpha$, thus

$$\alpha[z/x, y] \subseteq v(\alpha\langle z/x, y\rangle)$$

$\subseteq$) By definition, $fis_{xy}(\alpha)$ represents the same order variety as α and for all order variety β, $v(ent_{xy}(\beta)) \subseteq \beta$. Thus $v(ent_{xy}(fis_{xy}(\alpha))) \subseteq \alpha$. Then we again have that $v(S\langle z/x, y\rangle)_z * (x \parallel y) \subseteq \alpha$. Then by definition and as identification is monotonic we have

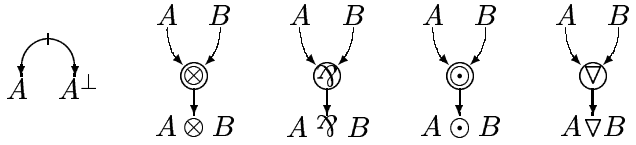
$$v(S\langle z/x, y\rangle)_z * z \subseteq \alpha[z/x, y] \qquad i.e. \qquad v(S\langle z/x, y\rangle) \subseteq \alpha[z/x, y]$$

□

4.4 MNL proof nets

We restrict us to the multiplicative fragment of NL i.e. to the formulae build from atoms $a, a^{\perp}, \ldots$, the commutative conjonction and disjonction (resp. $\otimes$ and $\parr$) and the non commutative conjonction and disjonction (resp. $\odot$ and ∇).

Definitions 15 (links and proof structures) *A* link *is an object for which the* premises *(input edges) and the* conclusions *(output edges) are two disjoint sets of vertices :*



A proof structure G *over the vertices* $V(G)$ *is a set of links such that :*
- *every vertex in* $V(G)$ *is a conclusion of (only) one link,*
- *every vertex in* $V(G)$ *either is a* conclusion of G *(i.e. is not a premise of any link of* G*) or is a premise of (only) one link,*
- *the set* γ *of the conclusions of* G *(writen* $G \vdash \gamma$*) is not empty.*

4.4.1 Maieli correctness criterion

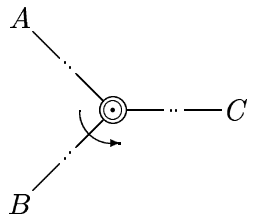
Definitions 16 (Switchings) *Let* G *a proof structure. A* switching s *for* G *is given by mutilating one premise-edge for each* ∇*-link and* $\parr$*-link. Any* ∇*-link (resp.* $\parr$*-link) admits a left/right mutilation wich is called the* left/right switch *of* ∇ *(resp.* $\parr$*). Any switching* s *for a proof structure* G *induces a graph on* $V(G)$ *which is called the* switched proof structure $s(G)$.

Fact 5 If a switched proof structure S induced by a proof structure $G \vdash \gamma$ is acyclic and connected then (viewing $\otimes$-nodes as $\circ$-nodes and $\odot$-nodes as $\bullet$-nodes, and effacing binary nodes implie that) S is a seaweed which represents a series-parallel order variety on γ.

Definition 17 (Suitable conclusion) *Let $G \vdash \gamma$ be a proof structure and s be a switching for G. Let a vertex of $s(G)$ labelled $A \nabla B$. A* conclusion suited to *$A \nabla B$ is a vertex $C \in \gamma$ such that there is no paths from $A \nabla B$ to C in $s(G)$ which is oriented in G.*

Definition 18 (M-correctness) *A proof structure G is* M-correct *iff for any switching s :*

1. *the switched proof structure $s(G)$ is acyclic and connected,*
2. *for any ∇-link labelled $A \nabla B$, for any suitable conclusion C, the intersection of the paths AB, AC and BC in the seaweed $s(G)$ is a $\odot$-node in G with the following anti-clockwise order :*



Theorem 10 ([Mai00]) *A proof structure G is M-correct iff G is sequentialisable.*

In the commutative fragment (multiplicative linear logic) the Maieli correctness criterion is exactly the Danos-Regnier's (the first step in the previous definition). The latter is well known to be in exponential time : if n is the number of $\parr$-links in a proof structure G then the Danos-Regnier correctness criterion checks 2^n graphs and cannot be inferred by the inspection of a fixed subset of the switches of G. So the Maieli correctness criterion is at least in exponential time.

4.4.2 The size of a proof structure

If we call *size* of a proof structure G the number of registers $size(G)$ required for the memorisation of G on some ramdom access machine (RAM) then in any non redundant coding, $size(G)$ is linear in the number of vertices of G i.e. $size(G) = \Theta(|V(G)|)$. Moreover, since the number of links in G is linear in the number of vertices of G, $size(G) = \Theta(|G|)$ also. In the following, one shall analyse the worst case asymptotic complexity of correctness in terms of $size(G)$.

Remark. It is usual to describe a proof net with only one conclusion : one only has to build a tree of $\parr$-links of the conclusions. One can see that this description does not improve the worst case asymptotic complexity.

4.5 Sequent calculus

Definition 19 *A* sequent $\vdash \alpha$ *consists of a series-parallel order variety* α *of formula occurences.*

Identity group	$\dfrac{}{\vdash A * A^{\perp}}$ (identity)	$\dfrac{\vdash \omega * A \quad \vdash \omega' * A^{\perp}}{\vdash \omega * \omega'}$ (cut)
Structural group	$\dfrac{\vdash \beta}{\vdash \alpha}$ (entropy), $\alpha \subseteq \beta$	
Logic group	$\dfrac{\vdash \omega * A \quad \vdash \omega' * B}{\vdash (\omega' < \omega) * A \odot B}$	$\dfrac{\vdash \omega * (A < B)}{\vdash \omega * A \nabla B}$
	$\dfrac{\vdash \omega * A \quad \vdash \omega' * B}{\vdash (\omega \parallel \omega') * A \otimes B}$	$\dfrac{\vdash \omega * (A \parallel B)}{\vdash \omega * A \parr B}$

We can have a sequent calculus without an explicit rule for entropy : only the $\parr$-rule need this rule. So we can substitute the entropy rule and the $\parr$-rule by the following one given in [AR00] :

$$\frac{\vdash \alpha[A, B]}{\vdash \alpha[A \parr B / A, B]} \ (\parr\star\text{-rule})$$

where $\alpha[A \parr B/A, B]$ is the identification of definition 13. Indeed in the multiplicative fragment the two versions are equivalent : by lemma 7, we have

- $\alpha[A \parr B/A, B]_{A \parr B} * (A \parallel B) \subseteq \alpha$, so entropy and $\parr$-rule can mimic the $\parr\star$-rule,
- $\omega * (A \parallel B) \subseteq \alpha$ implies $\omega * (A \parallel B) \subseteq \alpha[A \parr B/A, B]_{A \parr B} * (A \parallel B)$, so $\parr\star$-rule is an optimized version of $\parr$-rule where entropy has been minimized.

See [Rue00] for a detailled explaination and consequences of removing the entropy rule in the full NL.

4.6 Contractibility criterion

Definition 20 (Contracting proof structure) *A* contracting proof structure G *over the vertices* $V(G)$ *is a set of links and seaweeds such that :*

- *every vertex in* $V(G)$ *either is a conclusion of (only) one link or is an extremity of (only) one seaweed,*

- *every vertex in $V(G)$ either is a* conclusion of G *(i.e. is not a premise of any link of G) or is a premise of (only) one link or is an extremity of (only) one seaweed,*
- *the set γ of the conclusions of G (writen $G \vdash \gamma$) is not empty.*

We consider the following system of rewriting rules called *contraction rules* which is applied from contracting proof sub-structures to seaweeds :
- no rules for axiom-link, $\otimes$-link, $\odot$-link : an axiom-link is already a seaweed, a $\otimes$-link is viewed as a $\circ$-node and a $\odot$-link as $\bullet$-node,
- associativity rules, sequential rules and par rule of figure 4.1.

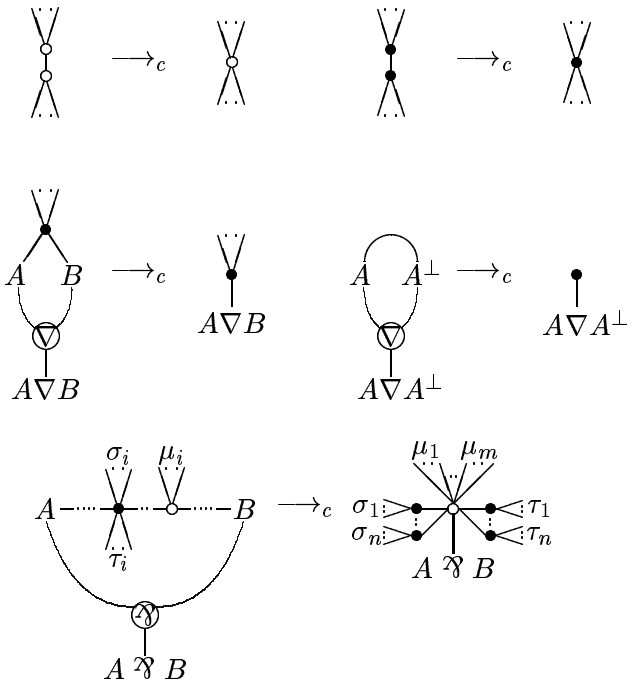


FIG. 4.1 – contraction rules

The par contraction rule corresponds to the transformation of a seaweed S and a $\parr$-link in $S\langle A \parr B/A, B\rangle$. We have $\mid n - m \mid \leq 1$ due to the alternate path between A and B.

Note that proof structures are particular contracting proof structures.

Definition 21 (Contractibility criterion) *A contracting proof structure G is c-correct if $\rightarrow_c^*$ reduces G to a seaweed.*

Theorem 11 (Confluence) *The system of contraction rules is confluent.*

Proof. There is no problems to do interactions with local rules like ∇-rule and associativity rules. The cases $\parr$-rule vs $\parr$-rule and $\parr$-rule vs ∇-rule are

treated in the appendix. The $\parr$-rule vs associativity rules are exactly the same as vs ∇-rule. □

Theorem 12 (Sequentialisation) *A proof structure G is c-correct iff G is sequentialisable.*

The proof can be deduced from the sequentialisation theorem from next section by using proposition 9.

Corollary 13 (Correctness) *A proof structure G is c-correct iff G is M-correct.*

This correctness criterion acts on an initial contracting proof structure G with $size(G)$ links and nodes of seaweeds (recall that axiom-links are seaweeds). Let $n = \Theta(size(G))$ be the sum of weighted number of links and the number of nodes. The analysis of each step of reduction shows that the number of links always decreases and that :
- the associativity decreases the number of nodes of the seaweed,
- the ∇-rule decreases the number of links without changing the number of nodes. In the degenerated case, to assign a weight of 2 to ∇-links allows to decrease n.
- the $\parr$-rule act on an alternate path. Let r and s be respectively the number of $\bullet$-nodes and $\circ$-nodes on this path. The contraction rule reduces the $r+s$ nodes to $2r+1$ nodes with $\mid r-s \mid \leq 1$ due to the alternate. Then in the worst case, the difference is of 2. So to assign a weight of 3 to $\parr$-links allows to decrease n.

So in the worst case (when G is c-correct), the number of steps of reduction in this criterion is linear in $size(G)$. Each step of reduction is a choice of a rule (it is linear time in $size(G)$) and the application of this rule. This decreases n down to 0.
Applying a reduction rule is linear in $size(G)$ in the worst case : the associativity rules and the ∇-rules are in constant time, the $\parr$-rule is linear in the length of the path. Indeed this latter rule consists of an $S\langle z/x, y\rangle$ operation of some A and B into $A \parr B$: this requires a linear time for fis_{AB} as well as for ent_{AB} and for ass_{AB}.
Therefore this correctness criterion is in quadratic time.

4.7 Parsing

In the previous section, we are dealing with contracting proof structure i.e. with seaweeds. Here is the same quadratic time parsing algorithm that checks the correctness of a proof structure but the objects are directly order varieties. From the sequent calculus one can find a non determinist algorithm for the sequentialisation of proof structures. We present here a determinist

reformulation. In order to show this, we introduce the parsing box which contains an order variety : let α an order variety on a set X,

is called the *parsing box* α. This a kind of link without premises which has one conclusion for each element of X. We use the following set of rules $\rightarrow_p$:

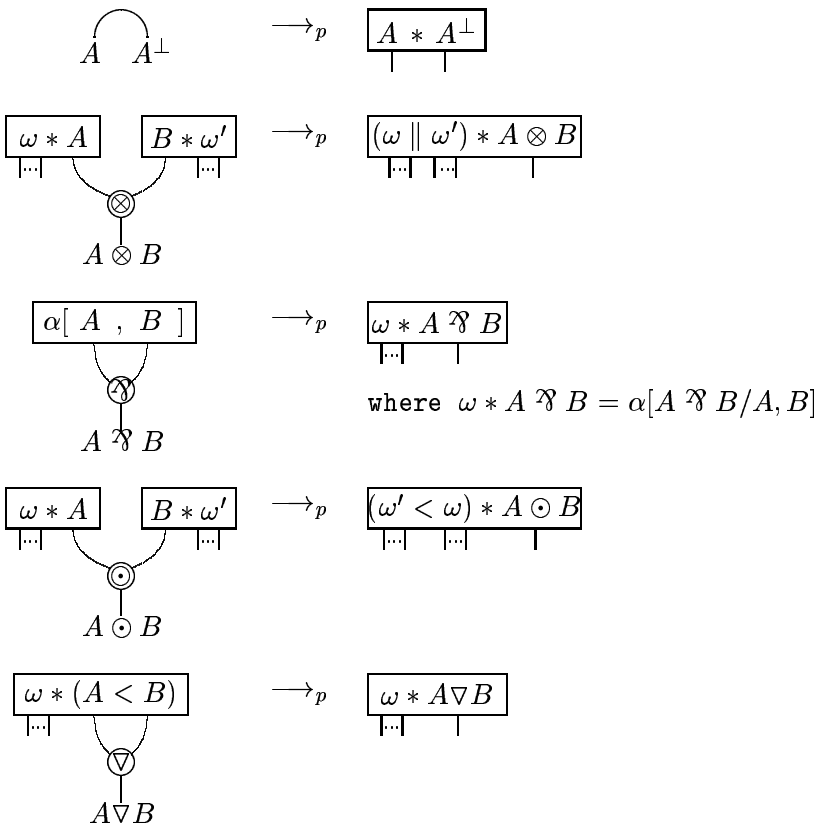


FIG. 4.2 – parsing rules

By the properties of $\rightarrow_c$ and proposition 9 we obtain the confluence of $\rightarrow_p$.

Lemma 14 *If* Π *is a proof in cut-free MNL of* $\vdash \alpha$ *then we can naturally associate with* Π *a proof net* Π^- *which reduces to the parsing box* $\beta \supseteq \alpha$.

Proof. The proof net Π^- is defined by induction on Π as follow :

Case 1 : Π is an axiom $\vdash A * A^\perp$; one must define Π^- as the axiom link : it is reduced to the parsing box $A * A^\perp$.

Case 2 : Π is obtain by a $\otimes$-rule from λ_1 and λ_2 which are respectively the proof of $\vdash \omega * A$ et $\vdash B * \omega'$; By induction hypothesis, λ_1^- and

λ_2^- are respectively reduced to the parsing boxes $\beta * A \supseteq \omega * A$ and $B * \beta' \supseteq B * \omega'$. Then we define Π^- as the tensor on A and B of λ_1 and λ_2 : it is reduced to the parsing box $(\beta \parallel \beta') * A \otimes B \supseteq (\omega \parallel \omega') * A \otimes B$.

Case 3 : Π is obtain by a $\parr$-rule from λ which is a proof of $\vdash \alpha[A, B]$; By induction hypothesis, λ is reduced to the parsing box $\beta[A, B] \supseteq \alpha[A, B]$. Then we define Π^- as the par on A and B of λ : it is reduced to the parsing box $\beta[A \parr B/A, B] \supseteq \alpha[A \parr B/A, B]$ by lemma 8.

Case 4 : Π is obtain by an entropy rule from λ which is a proof of $\vdash \beta$ with $\beta \supseteq \alpha$. Then we define Π^- as λ.

Case 5 : Π is obtain by a $\odot$-rule or a ∇-rule ; one can build Π^- like respectively in cases 2 and 3 if we recall that $\beta \subseteq \alpha[\omega < \omega']$ implies $\beta[\omega < \omega']$. □

Lemma 15 *If a proof net λ is reduced to the parsing box α then we can find a proof Π in sequent calculus of $\vdash \alpha$ such that $\Pi^- = \lambda$.*

Proof. By induction on the length of the reduction :

i) one step of reduction : λ is an axiom link which is reduced in the parsing box $A * A^\perp$. The claim is proved by taking as Π the axiom $\vdash A * A^\perp$.

ii) several steps of reduction : the system of parsing rules is confluent, so the last rule applied to λ is one of the followings :

- Tensor parsing rule : we have a proof net λ reduced in a parsing box $\beta = (\omega \parallel \omega') * A \otimes B$. So by the last step, there are the proof nets λ_1 and λ_2 reduced respectively in the parsing boxes $\omega * A$ and $B * \omega'$. By induction hypothesis, there is the proofs Π_1 and Π_2 in sequent calculus resp. of $\vdash \omega * A$ and $\vdash B * \omega'$ such that $\Pi_1^- = \lambda_1$ and $\Pi_2^- = \lambda_2$.
  So by taking as Π the tensor of $\vdash \omega * A$ and $\vdash B * \omega'$ we obtain a proof of $\vdash \beta$ such that $\Pi^- = \lambda$.
- Par parsing rule : we have a proof net λ reduced in a parsing box $\beta = \alpha[A \parr B/A, B]$. So by the last step, there is a proof net λ_1 reduced in a parsing box $\alpha[A, B]$. By induction hypothesis, there is a proof Π_1 in sequent calculus of $\vdash \alpha[A, B]$ such that $\Pi_1^- = \lambda_1$. One can take as Π the $\parr*$-rule of $\alpha[A, B]$ then $\Pi^- = \lambda$.
- The others parsing rules can be treated as in previous cases. □

Theorem 16 (Sequentialisation) *Let us say that the proof structure G is p-correct when $\rightarrow_p$ reduces G to a parsing box. Then, G is p-correct iff G is sequentialisable.*

Proof. Deduce from lemma 14 and 15. □

Corollary 17 *A proof structure G is p-correct iff G is M-correct.*

4.8 Conclusion

These criteria are like the others one from the cut management point of view. Given a sequent calculus proof of NL with cuts P, there is an associate proof net with cuts. The standard cut elimination give a cut-free proof net which can be sequentialised in a cut-free sequent calculus proof. Then this proof can be obtained from P by cut elimination. The question is to know what happens during the intermediate steps of cut elimination : is there a correctness criterion ? i.e. is there a sequentialisation theorem extended to proof nets with cuts ? In the commutative part of NL, the sequentialisation of proof nets with cuts can be solved by seeing a cut like a tensor for correctness. Detailed explanations can be found in [Laf95]. But these cannot be done here[2]. So how to deal with a contractibility correctness criterion for proof nets with cuts ?

The obtained correctness criterion is quadratic but there is a linear one for linear logic. This result comes from a reformulation of Danos contractibility criterion in terms of a sort of unification. This new Danos contractibility style criterion for NL is a first step in this direction.

4.9 Appendix : confluence proof

4.9.1 $\parr$-parsing rule v.s. $\parr$-parsing rule

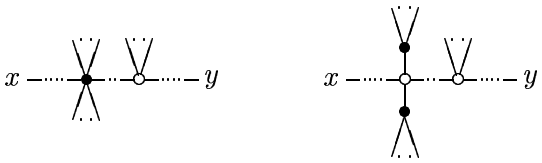
Let α be XSa order variety on a set X. We want to prove that for all distincts $A, B, C, D \in X$ we have $\alpha\langle A \parr B/A, B\rangle\langle C \parr D/C, D\rangle = \alpha\langle C \parr D/C, D\rangle\langle A \parr B/A, B\rangle$. In fact the $\alpha\langle x \parr y/x, y\rangle$ operation can be decompose in steps on sub-seaweeds : fis_{xy}, ent_{xy} and ass_{xy} are well defined for all nodes x, y in α. Note that the nodes x and y are not transformed in this processes : they are not in the open path between x and y (denoted $path(x, y)$).

We are only interrested in fis_{xy} and ent_{xy}. So for all x, y, z, t nodes in α we have the followings equations :

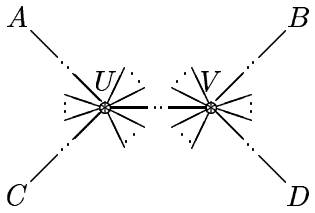
$$\begin{array}{lcll} ent_{xy}(fis_{xy}(S_{xy})) & = & T_{xy} & \\ ent_{xy}(fis_{xy}(T_{zt})) & = & T_{zt} & \\ ent_{xy}(fis_{xy}(S_{zt})) & = & S_{zt} & \text{if } path(x, y) \cap path(z, t) = \emptyset \end{array}$$

where we denote respectively by S_{xy} and T_{xy} the following forms of sub-seaweed of α which belong to the path between x and y :

[2] Solutions are given in [AR00] but they are not so elegant.



Let $A, B, C, D \in X$ and U, V two nodes in α such that $path(A, B) \cap path(C, D) = path(U, V)$. Then α is represented by :

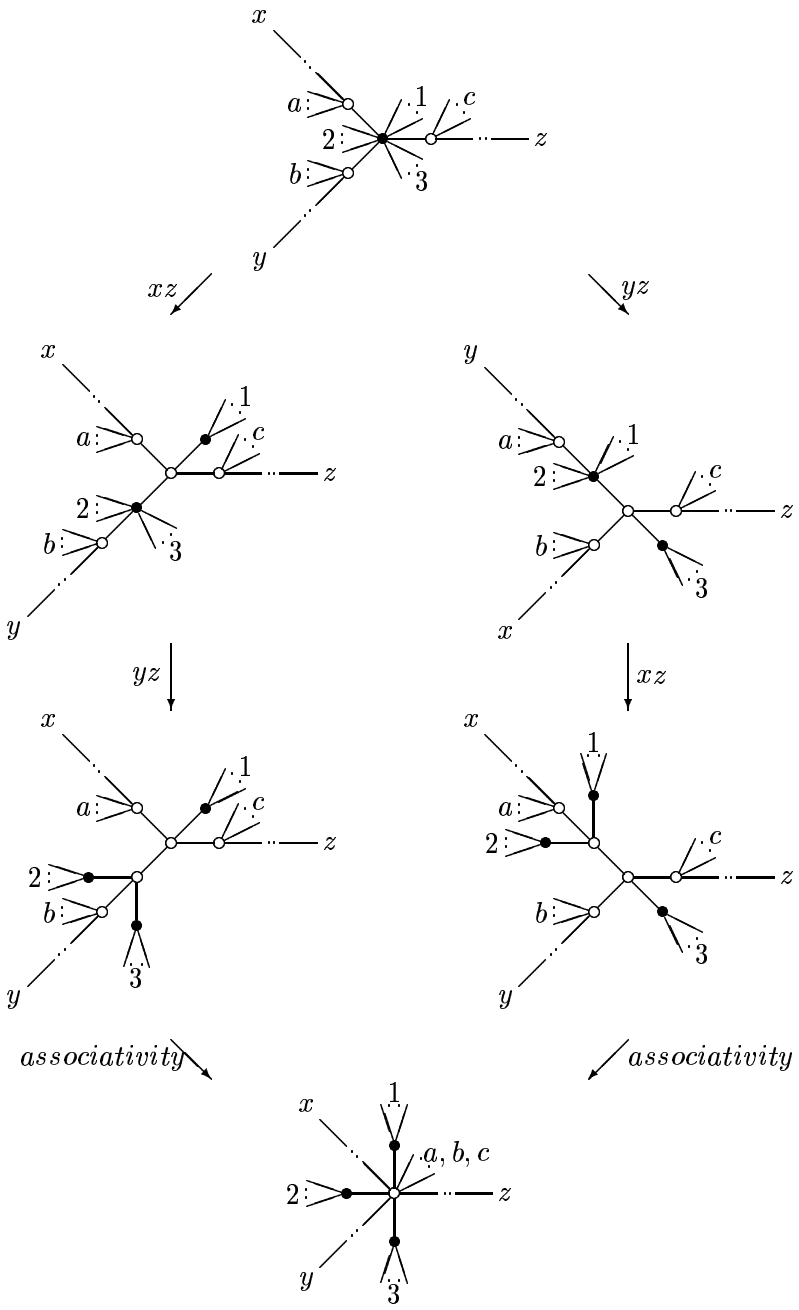


where U and V are undefined node. We have the following confluents diagrams :

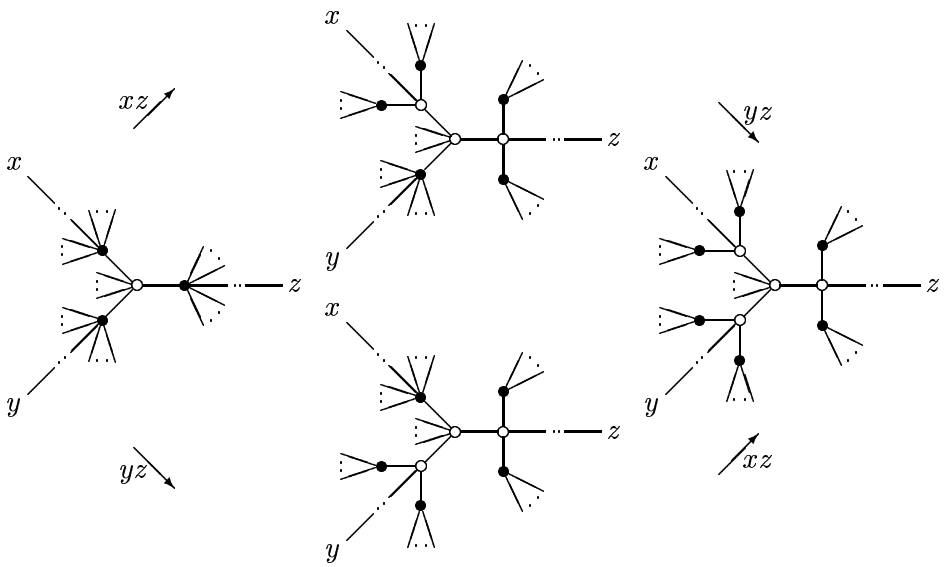
$$\begin{array}{ccc} S_{AU} & \xrightarrow{CD} & S_{AU} \\ \downarrow{\scriptstyle AB} & & \downarrow{\scriptstyle AB} \\ T_{AU} & \xrightarrow{CD} & T_{AU} \end{array} \qquad \begin{array}{ccc} S_{UV} & \xrightarrow{CD} & T_{UV} \\ \downarrow{\scriptstyle AB} & & \downarrow{\scriptstyle AB} \\ T_{UV} & \xrightarrow{CD} & T_{UV} \end{array}$$

where $\xrightarrow{xy}$ stands for $\xrightarrow{ent_{xy}(fis_{xy}(.))}$. And we have the same from S_{VB} (resp. S_{CU} and S_{VD}) to T_{VB} (resp. T_{CU} and T_{VD}).

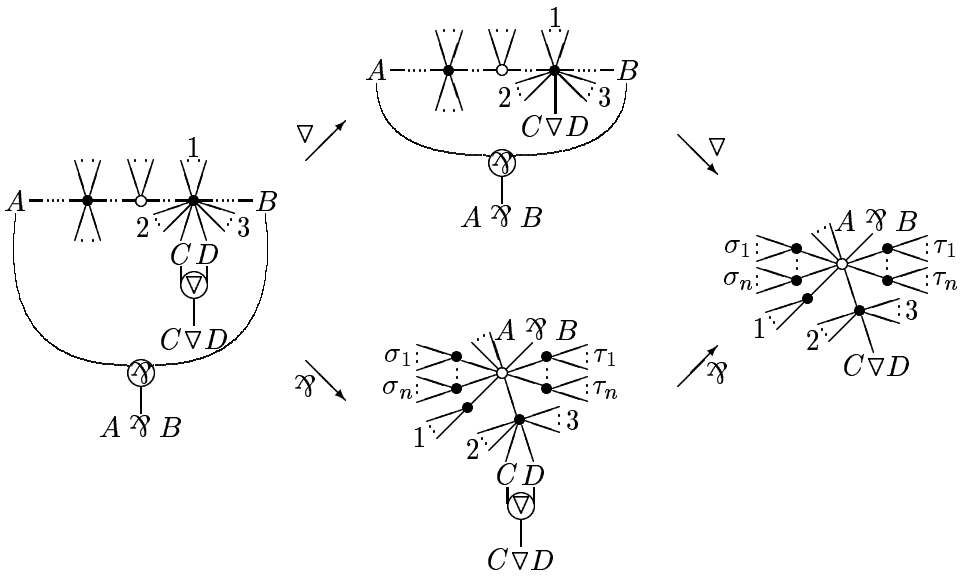
So what happends to U and V ? They can be treated in the same way : it depends only on the nature of the node.
- It is a serial node :



- It is a parallel node :



4.9.2 ⅋-parsing rule v.s. ▽-parsing rule



4.10 Critère pour les réseaux avec coupures

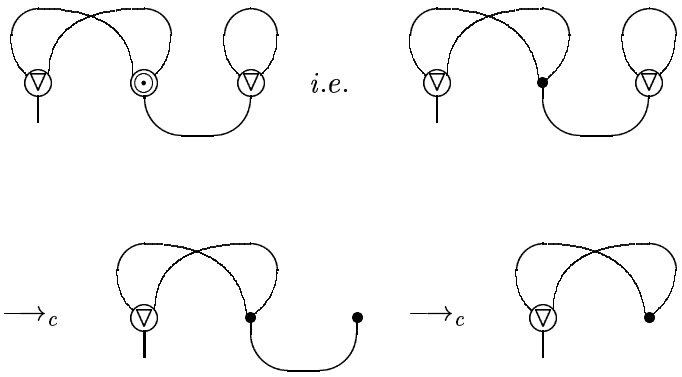
4.10.1 Règles de contraction

On définit le lien "coupure" par l'objet suivant ayant pour prémisses deux formules duales et pas de conclusion :

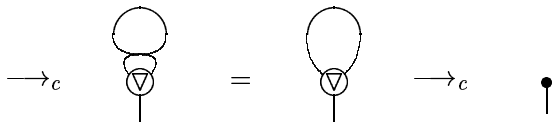
On étend les structures de preuve contractiles à celles construites avec les liens "axiome", les liens "tenseur", les liens "par", les liens "next", les liens "séquentiel" et les liens "coupure". On ajoute ainsi aux règles de contraction, que l'on applique à de telles structures de preuve pour donner des algues, l'idée suivante : il n'y a pas de règle de contraction pour les liens "coupure" car ce sont déjà des algues (comme le sont les liens "axiome"). On ajoute de façon explicite les règles suivantes permettant de traiter les cas de variétés d'ordre dégénérées (i.e. les variétés d'ordre vides sur un ensemble à un ou deux points)[3] :

On dira comme précédemment qu'une structure de preuve contractile (avec liens "coupure") est c-correcte si elle se contracte en une algue. Ce critère de contraction étendu reste donc confluent et quadratique.

Exemple 8



[3] les autres cas sur un ensemble à deux points sont tous des cas particuliers des règles générales. Il n'y en a pas d'autres sur un seul point car les connecteurs sont binaires.

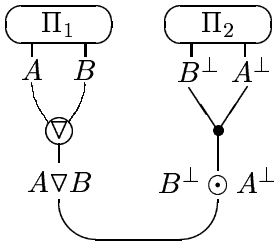


Lemme 25 (Acyclicité et connexité) *Si une structure de preuve (contractile) est c-correcte alors les structures de preuve (contractiles) avec interrupteurs correspondantes sont acycliques et connexes.*

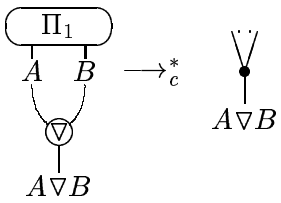
Preuve. Par l'absurde, si pour une position d'interrupteurs donnée ($\parr$ ou ∇) il existe un cycle alors il relie l'une des prémisses de l'interrupteur avec sa conclusion. Ce cycle existe toujours après utilisation d'une règle d'associativité, d'une règle de contraction ∇ ou d'une règle de contraction $\parr$. Ce qui contredit que la structure est c-correcte car une algue est acyclique. De même, si une position d'interrupteurs donnée déconnecte la structure alors on ne peut appliquer la règle de contraction correspondante. □

Lemme 26 (Elimination des coupures) *Une structure de preuve contractile obtenue après une étape de réduction depuis une structure c-correcte est c-correcte.*

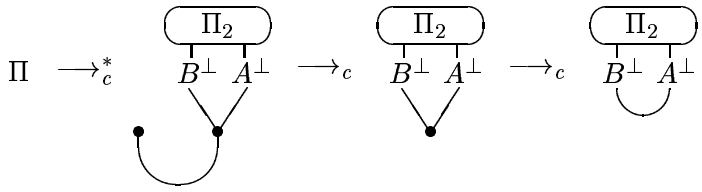
Preuve. Soit Π un réseau de preuve obtenu par coupure entre $A \nabla B$ et $B^\perp \odot A^\perp$. Or $\odot$ est un nœud non commutatif $\bullet$, donc comme Π est c-correct, d'après le lemme 25, $A^\perp$ et $B^\perp$ n'appartiennent pas à la même composante connexe que A dans la structure prémisse de la coupure. De même pour B, donc Π est de la forme



Par définition, comme Π est un réseau de preuve, Π_1 est c-correct et on a :

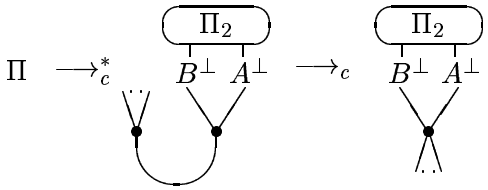


– Si on obtient une variété d'ordre vide alors



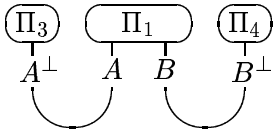
qui est c-correct. Donc d'après le lemme 25, Π_2 est formé de deux composantes connexes Π_3 et Π_4 séparant $A^\perp$ et $B^\perp$, chacunes étant des réseaux de preuve.

– Sinon on a alors



qui est c-correct. Donc Π_2 est de la même forme que précédemment.

Donc Π', le réduit de Π, est de la forme



Et on a la c-correction de Π'.

Le cas commutatif se traite de façon identique. □

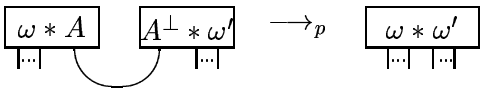
Théorème 27 (Adéquation et séquentialisation) *La c-correction est adéquate et séquentialisable relativement au calcul des séquents MNL avec coupures.*

La preuve peut être déduite d'un théorème d'adéquation et de séquentialisation pour un nouvel ensemble de règles de boites, à l'aide de la proposition 9. Il est décrit dans ce qui suit.

4.10.2 Règles de boites

On ajoute aux règles de boites[4] de la section 4.7 :

[4]Une boite désigne ce que l'on appelle en anglais une "parsing box". C'est la notion habituelle dans les réseaux de preuve : une boite est un objet dont on ne considére pas le contenu, mais qui peut avoir une information concernant sa frontière (ici la variété d'ordre de ses conclusions).

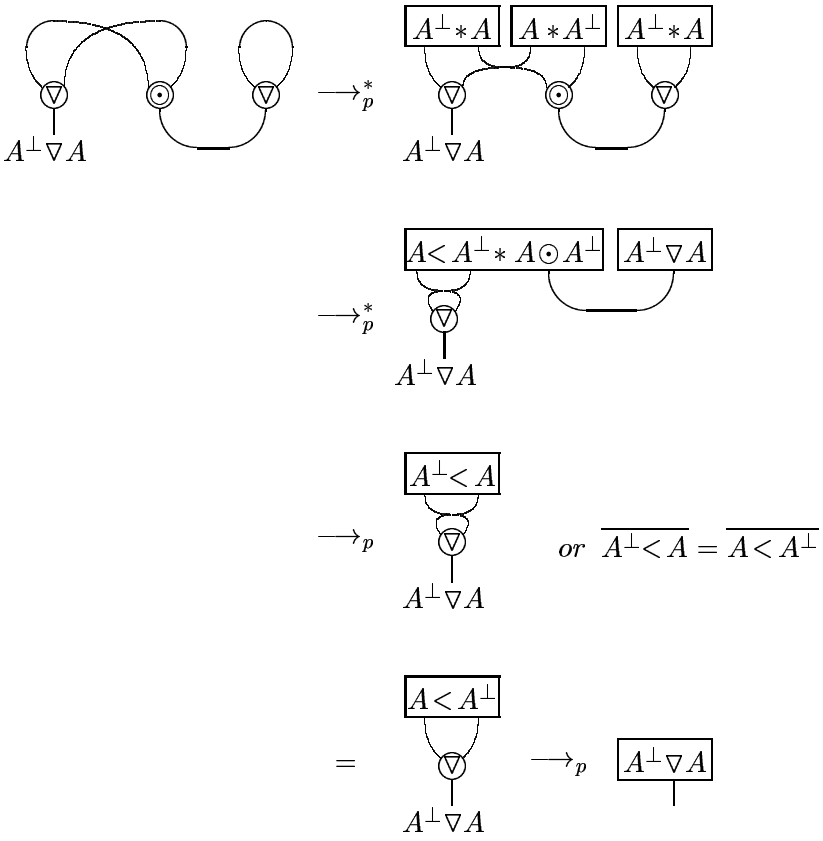


Ce critère de correction est l'analogue des règles de contraction. Il est ainsi quadratique. On dira qu'une structure de preuve est p-correcte si elle se réduit en une boite.

Discution sur la règle de coupure

On retrouve la préoccupation de gérer la règle de coupure dans les articles de M. Abrusci-P. Ruet [AR00] et R. Maieli [Mai00]. Les problèmes rencontrés par ces critères sont synthétisés dans les deux exemples qui suivent :

Exemple 9



Dans cet exemple, on constate que l'on a un réseau alors que l'une des deux composantes connexes des prémisses de la coupure n'est pas un réseau. C'est ce que l'on a dans le cas général. Au vue de cela il semble étrange que l'on puisse avoir un critère comme celui sur les boites, dont la seule règle traitant la coupure exprime le fait que les prémisses sont p-correctes (i.e. sont des

réseaux). En fait il se trouve que lorsqu'il y a une règle de coupure, l'une des prémisses peut correspondre à une variété d'ordre vide. Elle est cependant p-correcte. En calcul des séquents on a la règle suivante qui se décompose en une règle avec des variétés d'ordre non vide et des cas dégénérés de règles avec variétés d'ordre vides :

$$\frac{\vdash \omega * A \quad \vdash \omega' * A^{\perp}}{\vdash \omega * \omega'}$$

avec $|\omega| \geq 2$ *et* $|\omega'| \geq 2$

$$\frac{\vdash A \quad \vdash \omega' * A^{\perp}}{\vdash \overline{\omega'}} \qquad \frac{\vdash \omega * A \quad \vdash A^{\perp}}{\vdash \overline{\omega}}$$

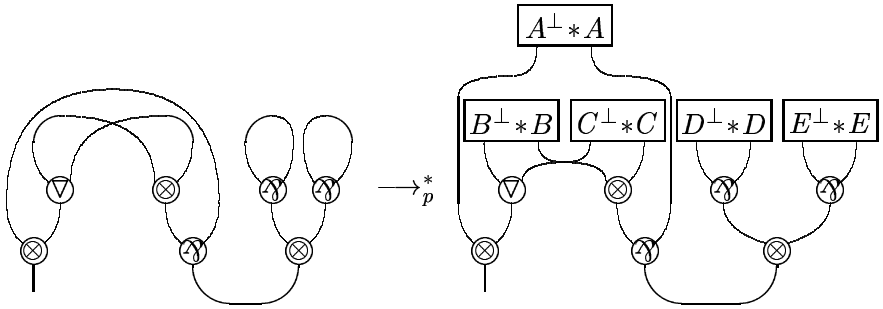
si $|\omega'| \geq 3$ *alors* $\overline{\omega'} \neq \emptyset$ *si* $|\omega| \geq 3$ *alors* $\overline{\omega} \neq \emptyset$

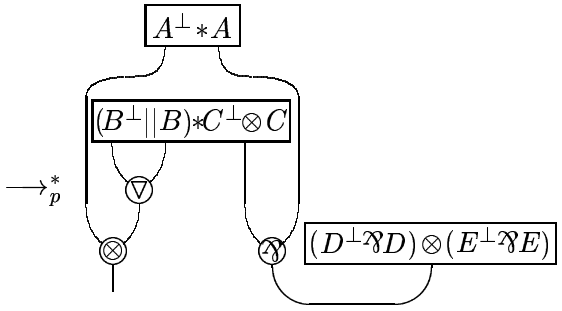
$$\frac{\vdash A \quad \vdash A^{\perp}}{\vdash}$$

Ces règles correspondent exactement aux différents cas de la règle de coupure de boites. On retrouve ces décompositions dans le critère de contraction : l'une des prémisses de la coupure est toujours une algue car c'est soit un lien $\otimes$ (i.e. $\circ$) soit un lien $\odot$ (i.e. $\bullet$). Dans le cas où celle-ci se contracte en une variété d'ordre vide, on ne peut alors appliquer les règles dégénérées ou d'associativités que si la seconde prémisse de la coupure est une algue. Donc que la prémisse conjonction (lien $\otimes$ ou $\odot$) d'une coupure soit dégénérée ou pas, il faut que la prémisse disjonction se contracte en une algue pour que la structure soit un réseau. Enfin, si la prémisse disjonction est contractable alors :

- si elle n'est pas dégénérée alors on retrouve le cas classique de coupure entre variétés d'ordre,
- sinon elle correspond à une variété d'ordre vide et les règles dégénérées de contraction propagent cette information à la prémisse conjonction. Le cas où cette dernière est une algue sous forme normale sur moins de quatre points continue la propagation. C'est le cas dans l'exemple 9.

Exemple 10 La structure suivante n'est pas séquentialisable. Cependant elle satisfait le critère de [AR00] (des solutions ont été proposées pour ne pas avoir ce type de problème).





Quelque soit le choix des littéraux A, B, C, D et E, on ne peut plus appliquer de règles $\rightarrow_p$. Donc cette structure n'est pas réductible en une boite (on peut montrer de même qu'elle n'est pas c-correcte). Ce n'est donc pas un réseau de preuve.

Théorème d'adéquation et de séquentialisation

La confluence du système de règles $\rightarrow_p$ est obtenue de celle de $\rightarrow_c$. On a ainsi les lemmes suivant correspondant aux lemmes 14 et 15 :

Lemme 28 *Si* Π *est une preuve de MNL (avec coupures) de* $\vdash \alpha$ *alors on peut naturellement associer à* Π *un réseau de preuve* Π^- *qui se réduit en la boite* $\beta \supseteq \alpha$.

Preuve. Le réseaux de preuve Π^- est défini par induction sur Π de la façon suivante :

Cas 1 : Π est un axiome ou est obtenu par une règle autre que celle de coupure. On définit alors Π^- de la même façon que dans le lemme 14.

Cas 2 : Π est obtenu par une règle de coupure sur la formule A entre λ_1 et λ_2. Ce sont des preuves respectives de $\vdash \omega * A$ et $\vdash A^\perp * \omega'$. Par hypothèse d'induction λ_1 et λ_2 sont réduits respectivement en les boites $\beta * A \supseteq \omega * A$ et $A^\perp * \beta' \supseteq A^\perp * \omega'$. Si on définit Π^- comme la coupure sur A entre λ_1 et λ_2 alors il est réduit par la règle de coupure de $\rightarrow_p$ en la boite $\beta * \beta' \supseteq \omega * \omega'$. □

Lemme 29 *Si un réseaux de preuve λ est réduit en une boite α alors on peut définir une preuve Π du calcul des séquents (avec coupures) de $\vdash \alpha$ telle que $\Pi^- = \lambda$.*

Preuve. Par induction sur la longueur de la réduction :

i) Une étape de réduction : on construit Π de la même façon que dans la preuve du lemme 15.

ii) Plusieures étapes de réduction : on regarde la dernière règle appliquée à λ. Soit ce n'est pas une règle de coupure, et alors on définit Π comme dans la preuve du lemme 15. Soit c'est une règle de coupure sur une formule A. On a alors un réseaux λ réduit en une boite $\beta = \omega * \omega'$. D'après la dernière règle, il existe λ_1 et λ_2 des réseaux de preuve se réduisant respectivement en les boites $\omega * A$ et $A^\perp * \omega'$. Par hypothèse d'induction, il existe Π_1 et Π_2 des preuves du calcul des séquents avec coupures respectivement de $\vdash \omega * A$ et $\vdash A^\perp * \omega'$ telles que $\Pi_1^- = \lambda_1$ et $\Pi_2^- = \lambda_2$. Dans le cas où la dernière règle est dégénérée, ω (ou ω') est l'ordre vide et alors Π_1 (ou resp. Π_2) est une preuve de $\vdash A$ (ou resp. $\vdash A^\perp$) satifaisant les propriétées.

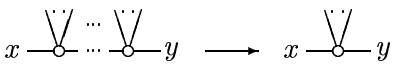
Donc si Π est définit comme la coupure sur A de $\vdash \omega * A$ et $\vdash A^\perp * \omega'$, on obtient une preuve de $\vdash \beta$ telle que $\Pi^- = \lambda$. □

Théorème 30 (Adéquation et séquentialisation) *Lap-correction est adéquate et séquentialisable relativement au calcul des séquents MNL avec coupures.*

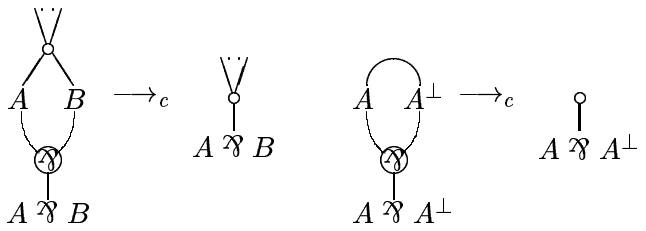
Preuve. Déduite des lemmes 28 et 29. □

4.11 Traduction de NL

La logique linéaire peut être obtenue par une *traduction commutative* de NL : on transforme chaque connecteurs $\odot$ en $\otimes$ et chaque connecteurs ∇ en $\parr$. Ainsi les nœuds non commutatifs $\bullet$ des algues sont traduits en nœuds commutatifs $\circ$. La règle d'associativité non commutative devient celle commutative. Un chemin entre deux feuilles x et y est alors décrit à l'aide de la figure suivante :



La règle de fission fis_{xy} (cf. section 14) suivit de la règle d'entropie ent_{xy} le long du chemin entre x et y laisse donc ce chemin inchangé. Faire de l'identification de x et y en z revient ainsi à simplement substituer z à $x||y$. La règle de contraction $\parr$ s'écrit donc de la façon suivante (avec son cas dégénéré) :



Ces règles sont exactement celles données par la traduction de la règle de contraction ∇. Ainsi les règles de contraction obtenues sont uniquement la règle d'associativité des nœuds $\circ$ et celles de contraction du $\parr$. Donc appliquer la traduction commutative de NL sur le critère de contraction donne le critère de contraction de V. Danos [Dan90].

Bibliographie

[Abr91] V. M. Abrusci. Phase semantics and sequent calculus for pure non-commutative classical linear propositional logic. *Journal of Symbolic Logic*, 56(4) :1403–1451, 1991.

[AR00] V. M. Abrusci and P. Ruet. Non commutative logic I : the multiplicative fragment. *Annals of Pure and Applied Logic*, 101 :29–64, 2000.

[Dan90] V. Danos. *Une application de la logique linéaire à l' étude des processus de normalisation (principalement de λ-calcul).* PhD thesis, Université Denis Diderot, Paris 7, 1990.

[DR89] V. Danos and L. Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28 :181–203, 1989.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50, 1987.

[Gir89] J.-Y. Girard. Towards a geometry of interaction. In John W. Gray and Andre Scedrov, editors, *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 69–108, Providence, Rhode Island, 1989. American Mathematical Society.

[Gue99] S. Guerrini. Correctness of multiplicative proof nets is linear. In *Fourteenth Annual IEEE Symposium on Logic in Computer Science*, pages 454–463. IEEE Computer Science, 1999.

[Laf95] Y. Lafont. From proof nets to interaction nets. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, pages 225–247. Cambridge University Press, 1995. Proceedings of the Workshop on Linear Logic, Ithaca, New York, June 1993.

[Lam58] J. Lambek. The mathematics of sentence structure. *Amer. Math. Mon.*, 65(3) :154–170, 1958.

[Mai00] R. Maieli. A new correctness criterion for multiplicative non-commutative proof-nets. Technical report, Institut de mathématiques de Luminy, 2000.

[MR00] R. Maieli and P. Ruet. Non commutative logic III : focusing proofs. Technical Report 2000-14, insitut de Mathématiques de Luminy, 2000.

[Rue00] P. Ruet. Non commutative logic II : sequent calculus and phase semantics. *Math. Struct. in Comp. Sci*, 10(2), 2000.

[Yet90] D. N. Yetter. Quantales and (noncommutative) linear logic. *Journal of Symbolic Logic*, 55(1) :41–64, 1990.

Index