UNIVERSITÉ AIX-MARSEILLE II - MEDITERRANÉE
U.F.R. DE SCIENCES

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ AIX-MARSEILLE II

Discipline : MATHÉMATIQUES

présentée et soutenue publiquement par

Olivier LAURENT

le 11 mars 2002

TITRE :

# ÉTUDE DE LA POLARISATION EN LOGIQUE

| | | | |
|---|---|---|---|
| Directeurs de thèse : | Jean-Yves | GIRARD | |
| | Laurent | REGNIER | |
| Rapporteurs : | Olivier | DANVY | |
| | Martin | HYLAND | |
| Jury : | Samson | ABRAMSKY | |
| | Pierre-Louis | CURIEN | |
| | Vincent | DANOS | |
| | Olivier | DANVY | |
| | Jean-Yves | GIRARD | |
| | Martin | HYLAND | |
| | Jean-Louis | KRIVINE | (Président) |
| | Laurent | REGNIER | |

*À mon grand-père.*

# Remerciements

Merci à mes deux anges-gardiens Vincent Danos et Laurent Regnier pour avoir veillé, entre Paris et Marseille, à l'avancée de mes travaux et à leur diffusion.
Merci à Jean-Yves Girard de m'avoir accueilli à Marseille, d'avoir accepté d'être mon directeur de thèse et de m'avoir laissé puiser abondamment dans ses idées. Merci à lui également pour faire de l'équipe LDP ce qu'elle est et pour la place importante qu'il insiste à donner aux thésards.
Merci à Samson Abramsky, Pierre-Louis Curien et Jean-Yves Girard de m'avoir apporté leur soutien inconditionnel lorsque c'était le plus important pour moi.

Merci à Martin Hyland, qui fut l'un des premiers à lire ce texte, pour les si agréables discussions que l'on a pu avoir lors de son séjour à Marseille et pour avoir accepté d'être rapporteur de cette thèse.
Merci à Olivier Danvy d'avoir accepté la lourde tâche d'apporter un regard informatique sur cette thèse et d'en être rapporteur.
Merci à Samson Abramsky de son intérêt pour mon travail et des encouragements qu'il m'a apportés.
Merci à Pierre-Louis Curien de l'attention qu'il a portée à mon travail et pour m'avoir encouragé à rejoindre son laboratoire.
Merci à Vincent Danos pour tout ce qu'il m'a appris de logique, son enthousiasme, son dynamisme scientifique, ...
Merci à Jean-Louis Krivine de m'avoir fait faire mes premiers pas en logique.
Merci à Samson Abramsky, Pierre-Louis Curien, Vincent Danos, Olivier Danvy, Jean-Yves Girard, Martin Hyland et Laurent Regnier d'avoir accepté de faire partie de mon jury et à Jean-Louis Krivine d'avoir accepté de le présider. Et merci à tous ces chercheurs actifs pour ce que leur travail peut apporter aux débutants comme encouragements à la recherche.

Merci à Myriam Quatrini et Lorenzo Tortora pour leur collaboration agréable et fructueuse.
Merci à Patrick Baillot, Russ Harmer et Peter Selinger pour toutes les discussions, capitales pour cette thèse, que nous avons pu avoir.
Merci à Patrick Baillot, Pierre Boudes, Alexandra Bruasse, Claudia Faggian, Sylvain Lippi, Virgile Mogbil et Samuel Tronçon d'avoir constitué une équipe de thésards de choc très serrée thématiquement ce qui a apporté une dynamique scientifique exceptionnelle. Leur participation à cette thèse a été évidente et quotidienne.
Merci à tous les membres de l'équipe LDP pour leur accueil, le plaisir que j'ai pu avoir à travailler avec eux et le soutien qu'ils m'ont apporté.
Merci aux membres de PPS de m'avoir accueilli pendant cette dernière année de thèse.
Merci à France Bodin, Jeanine Brohan et Aurélia et Éric Lozingot pour leur permanente disponibilité et le nombre de problèmes qu'ils m'ont si efficacement aidé à résoudre. Merci également à Pierre Barthélémy, Jean-Bruno Erismann et Pascal Rage pour leur aide. Et merci à eux tous pour leur dynamisme sans lequel l'IML ne serait pas ce qu'il est.

Merci à Laurent Regnier pour la qualité absolument exceptionnelle de son encadrement, tant sur le plan scientifique que personnel. Cette thèse lui doit énormément, bien au-delà de ce que l'on peut attendre d'un directeur de thèse.

Et merci à Anne pour tout le reste.

# Table des matières

# Introduction

La *théorie de la démonstration* est la branche de la logique qui s'intéresse à la structure des preuves mathématiques. L'un des résultats essentiels de ce domaine est l'*élimination des coupures*, qui montre que toute preuve peut être transformée en une preuve élémentaire, mais sans ingéniosité donc généralement incompréhensible. De son côté, l'*informatique fondamentale* s'intéresse aux différents paradigmes de programmation, aux langages, aux programmes et à leur évaluation.

L'objet de la *logique de la programmation* est l'étude des liens entre *calcul* du point de vue informatique et *structure géométrique des preuves* en logique. La clef d'une telle analyse est la possibilité d'interpréter l'élimination des coupures des systèmes logiques comme l'évaluation dans certains langages de programmation.

On cherche à établir le rapport entre l'exécution des programmes dans une classe donnée de langages et le processus de réduction des preuves d'un certain système logique. La bonne compréhension mathématique du côté logique permet alors d'analyser le comportement des programmes, inversement l'informatique suggère les directions à suivre en logique pour obtenir des systèmes pertinents. Un tel système doit en particulier être suffisamment expressif pour avoir un intérêt informatique et suffisamment bien compris pour apporter une analyse sur le calcul. L'éclairage réciproque des deux versants de cette correspondance est d'autant plus riche que le lien entre les deux est précis et bien analysé.

Historiquement, le premier système logique ayant une notion d'élimination des coupures est le système de logique classique $\mathsf{LK}$ de Gentzen[1]. Dans ce premier cas, il s'agit d'un résultat d'existence : «il existe une manière d'éliminer les coupures». L'élimination des coupures définit une transformation des preuves dont on sait qu'elle termine et qu'elle permet d'associer à toute preuve, une preuve «élémentaire» (sans coupure) mais ce résultat n'est a priori pas unique. Le processus de calcul que ceci permet d'interpréter est donc fortement non déterministe.

L'interprétation calculatoire de la logique, ou l'interprétation logique du calcul, a en fait réellement débuté avec la logique intuitionniste (minimale) $\mathsf{LJ}$ à travers la désormais célèbre *correspondance de Curry-Howard* qui explicite le lien entre l'élimination des coupures dans $\mathsf{LJ}$ et la réduction du $\lambda$-calcul (programmation fonctionnelle). On obtient cette fois une analogie précise dans un cadre déterministe. L'élimination des coupures est alors un processus qui termine et qui mène à l'unique preuve sans coupure possible.

L'existence de traductions, dites $\neg\neg$-traductions, de la logique classique dans la logique intuitionniste donne un moyen d'analyser également la logique classique. Cependant, il s'agit d'un biais indirect qui ne donne pas d'interprétation immédiate des preuves classiques. Ce n'est que beaucoup plus tard que la possibilité d'interpréter directement la logique classique dans un cadre calculatoire déterministe est apparue. Il s'avère que les instructions de contrôle (`call/cc`, exceptions, ...) correspondent à des formules classiques qui ne sont pas prouvables intuitionnistiquement (loi de Peirce,

[1] Les notations des systèmes logiques considérés sont définies à l'annexe A.1 page 213.

raisonnement par l'absurde, tiers exclu, ...). Cette correspondance a donné naissance en logique à différents systèmes pour la logique classique (c'est-à-dire ayant la même prouvabilité que LK) possédant une élimination déterministe des coupures, ce qui donne des extensions de LJ dans lesquelles on peut interpréter les langages fonctionnels avec primitives de contrôle.

Parallèlement à cela est apparue la logique linéaire LL, qui permet en particulier de raffiner l'étude de la logique intuitionniste, et donc de préciser l'analyse des langages fonctionnels. Des travaux ont également été faits pour interpréter la logique classique dans LL mais les traductions sont complexes et moins informatives (elles cachent en réalité des $\neg\neg$-traductions). De plus les rapports entre logiques classique et intuitionniste disparaissent puisque ces deux systèmes sont plongés dans un seul et même système : LL.

À la complexité de la traduction de la logique classique dans LL, s'ajoute la difficulté d'étudier la logique linéaire dans son ensemble. Ce système très riche est complexe à analyser et de nombreuses directions d'étude restent à ce jour limitées à des sous-systèmes de LL. Il est souvent nécessaire de se restreindre à des fragments plus faciles à étudier mais encore faut-il qu'ils soient suffisamment expressifs pour pouvoir y traduire les traits de programmation que l'on souhaite analyser. Ainsi, afin d'étudier LJ sans pour autant devoir affronter le système LL dans son ensemble, il est possible de se restreindre à la logique linéaire intuitionniste ILL qui, tout en contenant la traduction de LJ (donc des langages fonctionnels), a permis de meilleures avancées dans certains domaines (réseaux de preuve, sémantique des jeux, ...).

L'objectif de cette thèse est d'étendre cette approche à la logique classique : peut-on caractériser un sous-système de LL (ou une variante) interprétant de manière directe la logique classique (donc les langages fonctionnels avec primitives de contrôle) et possédant des propriétés plus simples que LL ?

Notre approche est basée sur la notion de *polarités* qui, bien que déjà implicitement présente en logique intuitionniste, est véritablement issue d'une analyse fine des propriétés de la logique linéaire. Cette idée a été explicitement introduite par Girard pour son système de logique classique déterministe LC et sert de clef de voûte à son programme actuel de Ludique. Partant de LC, il est naturel de définir dans LL deux classes de formules (les formules *positives* et les formules *négatives*) échangées par la négation et basées sur une séparation en deux classes des connecteurs de la logique linéaire. On est alors amené à introduire la *logique linéaire polarisée* LLP. Il reste à voir si ce système permet de mieux répondre au problème de l'analyse de la logique classique que LL ou ILL, ce qui revient à aborder les trois questions suivantes qui constituent la colonne vertébrale de cette thèse :
- Quelles sont les propriétés de LLP et sont-elles plus simples que celles de LL ?
- Quelle est l'expressivité de LLP ? est-elle suffisante pour la logique classique et les langages fonctionnels avec contrôle ?
- Est-il possible d'étudier, au niveau linéaire, les rapports entre logiques classique et intuitionniste ?

Bien que né de la contrainte de LL par polarisation, LLP apparaît comme un système logique à part entière, plus structuré que LL, simplifiant notablement sa théorie de la démonstration (réseaux de preuve polarisés, jeux polarisés, ...) et donc l'étude de son élimination des coupures.

De plus, son pouvoir expressif lui permet d'étendre conservativement la relation entre LJ et LL dont l'analyse a permis des résultats remarquables en théorie des langages fonctionnels (sémantique dénotationnelle, sémantique des jeux, ...) en une relation analogue (mais classique) entre LK et LLP. L'image de LJ dans LLP est alors facilement délimitable ($\mathsf{LL}_{\text{pol}}$) et le rapport qu'elle entretient avec LLP permet d'étudier celui de LJ avec LK.

Contrairement au cas général de LL, pour lequel il est nécessaire de considérer des fragments pour l'étude de certaines notions, nous allons effectuer une étude exhaustive de la théorie de la démonstration de LLP, tant sur le plan syntaxique que sémantique. On peut montrer que toutes les branches usuelles de l'étude de la logique linéaire peuvent être appliquées à LLP dans son intégralité. Ce sera l'objet de la partie II après des rappels sur quelques notions importantes de LL dans la partie I (chapitres 1, 2 et 3).

Dans le chapitre 4, nous précisons comment se fait la construction de LLP à partir de LL et de l'idée de polarisation. Puis, nous étudions les propriétés logiques fondamentales de LLP à l'aide du calcul des séquents.

Dans le chapitre 5, nous abordons la principale syntaxe utilisée pour LL : les réseaux de preuve. C'est la principale nouveauté due à LL, en supprimant des informations de séquentialité inutiles du calcul des séquents elle clarifie grandement la procédure d'élimination des coupures grâce à la diminution du nombre d'étapes à considérer. Contrairement au cas de LL, nous définissons des réseaux pour LLP tout entier avec une procédure complète d'élimination des coupures. De plus les polarités permettent de donner un critère de correction nettement plus simple que pour LL. Nous abordons les trois principales notions de réseaux avec connecteurs additifs : avec boîtes, avec poids et par tranches. L'idée de réseaux par tranches a été avancée par Girard mais n'a jamais pu être réellement appliquée jusqu'ici.

Dans le chapitre 6, nous utilisons les espaces de corrélation et les catégories de contrôle pour définir les premières sémantiques dénotationnelles de LLP. Ceci place LLP dans le contexte des études précédentes puisque ces deux notions préexistaient à LLP et fournissent des modèles.

Dans le chapitre 7, la sémantique des jeux usuelle (jeux intuitionnistes) est étendue à une véritable dualité Joueur/Opposant (correspondant à positif/négatif) permettant de définir un modèle de jeux pour LLP. Ces jeux polarisés contiennent les jeux intuitionnistes et clarifient les relations entre les différentes constructions par l'introduction d'*opérateurs de décalage*.

Dans le chapitre 8, après avoir donné une interprétation des connecteurs additifs en géométrie de l'interaction (GoI) qui simplifie celle de Girard, nous étendons la GoI aux règles structurelles généralisées de LLP grâce à une machine parallèle.

Afin de montrer l'expressivité de LLP, nous montrerons comment y interpréter les principaux systèmes de logique classique déterministes connus, aussi bien pour l'appel par nom que pour l'appel par valeur. L'étude de ces traductions constitue la partie III.

Dans le chapitre 9, nous décrivons le cœur des trois principales traductions des systèmes classiques dans LLP : la traduction polarisée, la traduction positive (correspondant à l'appel par nom) et la traduction négative (correspondant à l'appel par valeur). Ce qui permet, en particulier, une classification précise des traductions déjà connues.

Dans le chapitre 10, nous appliquons la traduction polarisée aux deux principaux systèmes classiques polarisés : LC et $\mathsf{LK}^{\eta}_{\mathrm{pol}}$.

Dans le chapitre 11, nous étudions en détails les liens entre LLP et le $\lambda\mu$-calcul, aussi bien pour l'appel par nom que pour l'appel par valeur ainsi que les liens avec deux autres extensions classiques du $\lambda$-calcul. Comme pour l'interprétation du $\lambda$-calcul dans LL, l'interprétation du $\lambda\mu$-calcul dans LLP nous donne des retombées sur le $\lambda\mu$-calcul lui-même : la $\sigma$-équivalence.

Dans le chapitre 12, nous terminons les traductions de systèmes classiques dans LLP par deux cas de systèmes non déterministes possédant une version déterminisée par nom et une par valeur. On a ainsi les sous-systèmes déterministes LKQ (appel par valeur) et LKT (appel par nom) de $\mathsf{LK}^{\mathsf{tq}}$ et le $\bar{\lambda}\mu\tilde{\mu}_Q$-calcul (par valeur) et le $\bar{\lambda}\mu\tilde{\mu}_T$-calcul (par nom) pour le $\bar{\lambda}\mu\tilde{\mu}$-calcul.

Ces deux parties répondent ainsi aux deux premières questions que nous nous sommes posées.

La partie IV s'intéresse à ce que l'analyse dans LLP apporte sur les systèmes classiques que l'on a traduits (on a vu un premier exemple avec la $\sigma$-équivalence).

Dans le chapitre 13, on exploite l'idée que LLP est à LL (ou en tout cas son fragment polarisé $\mathsf{LL}_{\mathrm{pol}}$) ce que LK est à LJ permettant une analyse linéaire du rapport entre classique et intuitionniste. Ceci se traduit concrètement en montrant que les différentes traductions de LLP dans LL que l'on a rencontrées sont des analogues linéaires des $\neg\neg$-traductions de LK dans LJ (avec certaines optimisations).

Dans le chapitre 14, nous étudions le sous-système linéaire MALLP de LLP, un système plus structuré que le fragment linéaire MALL de LL. Nous montrons que MALLP correspond précisément à d'autres fragments linéaires de systèmes classiques ($\lambda\mu$-calcul linéaire par exemple), ce qui fait apparaître l'idée d'une *logique classique linéaire*. C'est une idée nouvelle dans la mesure où la logique classique est généralement considérée comme caractérisée par l'utilisation des règles structurelles sur toutes les formules. Il apparaît ici que ce n'est pas nécessairement la caractéristique principale de la logique classique.

Pour conclure, nous montrons dans le chapitre 15 comment LLP permet d'analyser la dualité entre appel par valeur et appel par nom. Cette dualité est réapparue récemment dans différents travaux et s'exprime de manière particulièrement claire dans LLP par la dualité entre positif et négatif.

# Première partie

# Rappels

Comme son titre l'indique, cette partie ne contient ni définitions ni résultats nouveaux, il s'agit simplement de rappeler un certain nombre de définitions et de résultats dans le cadre général de la logique linéaire avant de les appliquer au cas polarisé.

Il faut toutefois préciser que la présentation faite dans le chapitre 3 est originale sur différents points bien que très fortement inspirée de celle de la thèse de G. McCusker [McC96]. Elle met en avant certaines définitions qui, bien qu'utilisées dans la littérature, n'étaient pas autant mises en valeur, en particulier en ce qui concerne l'utilisation des jeux indépendemment des stratégies à travers les morphismes d'arènes et de parties.

# Chapitre 1

# Réseaux de preuve

Nous rappelons dans ce chapitre les principaux résultats sur les réseaux de preuve pour MELL *avec quantificateurs du second ordre* et *sans constante* [Gir87, Gir91b, Dan90, Reg92]. Les réseaux de preuve constituent la principale nouveauté syntaxique de la logique linéaire. Inspirés de la déduction naturelle et utilisant la dualité de LL, ils permettent une représentation des preuves par des graphes ce qui fournit une structure moins séquentielle que les arbres du calcul des séquents. Il en découle que l'étude de l'élimination des coupures est nettement simplifiée puisque que la plupart des étapes commutatives nécessaires en calcul des séquents disparaissent.

L'intégration des connecteurs additifs dans les réseaux de preuve n'ayant à ce jour pas de réponse pleinement satisfaisante, nous aborderons plus tard cette question, dans le cadre polarisé, où nous comparerons les diverses solutions existantes. Il en va de même pour les constantes.

## 1.1 Structures de preuve

1.1. Définition (Structures de preuve)
Une *structure de preuve* est un graphe orienté acyclique à arêtes *pendantes*, *i.e.* dont certaines arêtes ont un nœud source mais pas de nœud cible (ces arêtes sont appelées *conclusions* de la structure de preuve). Les *nœuds* (ou *liens*) sont étiquetés par l'un des symboles *ax*, *cut*, $\parr$, $\otimes$, $!$, $?d$, $?c$, $?w$, $\forall$ ou $\exists$ (et correspondent aux règles du calcul des séquents). Les arêtes sont étiquetées par des formules de LL appelées *type* de l'arête. L'étiquette de chaque nœud impose son nombre d'arêtes entrantes (ses *prémisses*) et sortantes (ses *conclusions*) ainsi que des relations entre les types de ces arêtes, comme indiqué sur la figure 1.1. Par exemple un nœud $\otimes$ a deux arêtes entrantes de types (arbitraires) $A$ et $B$ et une arête sortante dont le type doit être $A \otimes B$.

À chaque nœud ! de prémisse $A$ d'une structure de preuve est associée une sous-structure de preuve (appelée *boîte* du nœud) de conclusions $?\Gamma, A$. La conclusion $A$ est appelée *porte principale* et les autres sont les *portes auxiliaires*. Deux boîtes doivent toujours être soit disjointes soit incluses l'une dans l'autre. On appelle *profondeur* d'un nœud le nombre de boîtes qui le contiennent.

Soit $n$ un nœud $\forall$ dont la conclusion a type $\forall X A$, on appelle $X$ la *variable propre* de $n$. Les variables propres de deux liens $\forall$ doivent être distinctes et les conclusions de la structure de preuve ne doivent pas contenir de variables propres libres. Si $n$ est à l'intérieur d'une boîte, sa variable propre ne doit pas apparaître hors de cette boîte.

Si la conclusion d'un nœud contient, comme variable libre, la variable propre $X$ d'un lien $\forall$, on dit qu'il *dépend* de $X$.

Notations (Représentation et nomenclature) : Les structures de preuve seront toujours re-

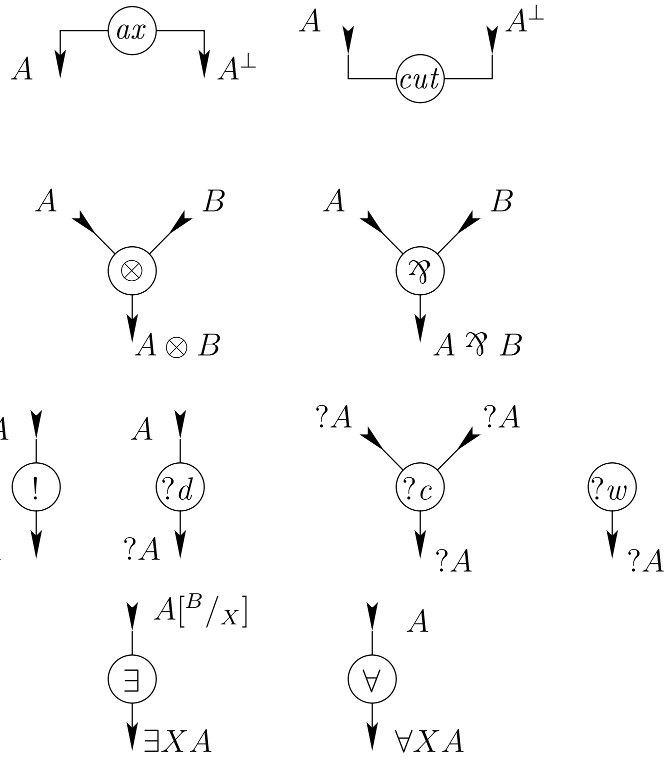



FIG. 1.1 – Nœuds des structures de preuve pour MELL

présentées avec les arêtes orientées du haut vers le bas ce qui permet d'omettre les orientations. Du fait de cette convention on parlera de nœud (ou d'arête) situé *au-dessus* ou *au-dessous* d'un autre, de *haut* ou de *bas* d'une structure de preuve, de *monter* ou *descendre* dans une structure de preuve, ...

Un nœud à profondeur 0 dont les conclusions sont des arêtes pendantes est appelé *nœud conclusion*. Les nœuds conclusions et les nœuds *cut* à profondeur 0 sont appelés *nœuds terminaux*.

1.2. DÉFINITION (TAILLE)
La *taille* d'une structure de preuve est son nombre de nœuds.

Nous allons maintenant rappeler la traduction des preuves du calcul des séquents MELL dans les structures de preuve.

1.3. DÉFINITION (TRADUCTION DU CALCUL DES SÉQUENTS)
Soit $\pi$ une preuve de $\vdash \Gamma$ en calcul des séquents, on lui associe la structure de preuve $\mathcal{S}_\pi$ de conclusions $\Gamma$, définie inductivement à partir de $\pi$ en regardant la dernière règle $R$ (on note $\pi_1$ ou $\pi_1$ et $\pi_2$ les preuves des prémisses de $R$ selon qu'elle est d'arité 1 ou 2) :

(*ax*) $\mathcal{S}_\pi$ contient un unique lien *ax*.

(*cut*) On met un lien *cut* entre les conclusions de $\mathcal{S}_{\pi_1}$ et $\mathcal{S}_{\pi_2}$ qui correspondent aux formules coupées dans $R$.

($\otimes$) On met un lien $\otimes$ entre les conclusions de $\mathcal{S}_{\pi_1}$ et $\mathcal{S}_{\pi_2}$ qui correspondent aux formules actives de $R$.

($⅋$) On met un lien $⅋$ entre les conclusions de $\mathcal{S}_{\pi_1}$ qui correspondent aux formules actives de $R$.

(!) On ajoute à $\mathcal{S}_{\pi_1}$ un lien ! sous la conclusion correspondant à la formule active et on lui associe $\mathcal{S}_{\pi_1}$ comme boîte.

($?d$) On met un lien $?d$ sous la conclusion de $\mathcal{S}_{\pi_1}$ qui correspond à la formule active de $R$.

($?c$) On met un lien $?c$ entre les conclusions de $\mathcal{S}_{\pi_1}$ qui correspondent aux formules actives de $R$.

($?w$) On ajoute à $\mathcal{S}_{\pi_1}$ un lien $?w$ avec comme conclusion la formule affaiblie dans $R$.

($\forall$) On ajoute à $\mathcal{S}_{\pi_1}$ un lien $\forall$ sous la conclusion correspondant à la formule active dans $R$. L'absence de la variable propre dans le contexte de la règle garantit qu'elle n'apparaît pas libre dans les conclusions de la structure de preuve.

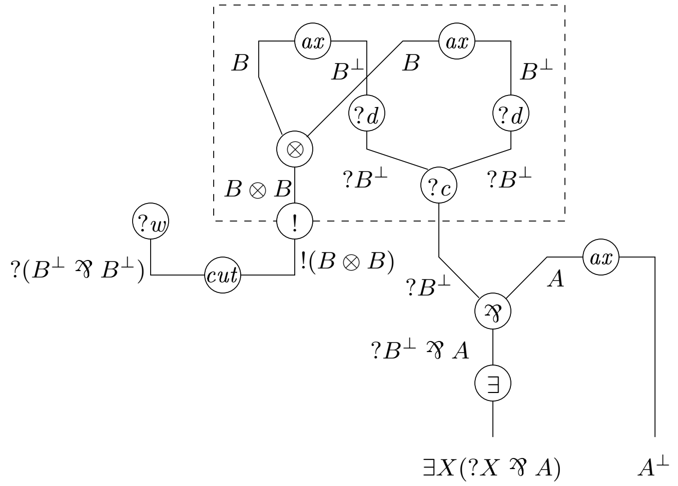
($\exists$) On ajoute à $\mathcal{S}_{\pi_1}$ un lien $\exists$ sous la conclusion correspondant à la formule active dans $R$.

1.4. Exemple

La preuve du calcul des séquents :

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash B, B^\perp$}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash B, B^\perp$}
\RightLabel{$?d$}
\UnaryInfC{$\vdash B, ?B^\perp$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash B \otimes B, B^\perp, ?B^\perp$}
\RightLabel{$?d$}
\UnaryInfC{$\vdash B \otimes B, ?B^\perp, ?B^\perp$}
\RightLabel{$?c$}
\UnaryInfC{$\vdash B \otimes B, ?B^\perp$}
\RightLabel{$!$}
\UnaryInfC{$\vdash !(B \otimes B), ?B^\perp$}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash A, A^\perp$}
\RightLabel{$?w$}
\UnaryInfC{$\vdash ?(B^\perp \parr B^\perp), A, A^\perp$}
\RightLabel{$cut$}
\BinaryInfC{$\vdash ?B^\perp, A, A^\perp$}
\RightLabel{$\parr$}
\UnaryInfC{$\vdash ?B^\perp \parr A, A^\perp$}
\RightLabel{$\exists$}
\UnaryInfC{$\vdash \exists X(?X \parr A), A^\perp$}
\end{prooftree}
$$

donne le réseau de preuve suivant :

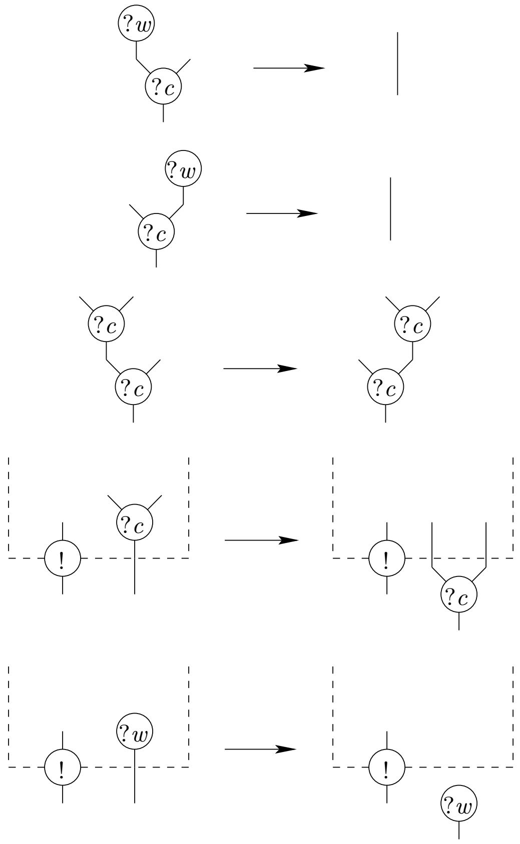



1.5. Définition (Arbre exponentiel)

Soit $\mathcal{S}$ une structure de preuve et $a$ une arête de $\mathcal{S}$ de type $?A$, l'*arbre exponentiel* de $a$ est l'arbre des nœuds $?c$ et $?w$ (dont les arêtes ont type $?A$) situés au-dessus de $a$ dans $\mathcal{S}$.

1.6. Définition (Équivalence structurelle)

L'*équivalence structurelle* $\simeq_s$ de structures de preuve consiste à modifier la structure des *arbres exponentiels*. Elle est définie comme la clôture réflexive symétrique transitive de la relation de réécriture $\rightarrow_s$ :




## 1.2 Critère de correction

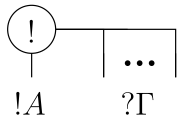
Afin de caractériser les structures de preuve qui correspondent à des preuves du calcul des séquent, on est amené à introduire la notion de *critère de correction*. Il s'agit d'une condition à vérifier sur une structure de preuve pour s'assurer qu'elle peut être transformée en preuve (séquentialisée). Ce critère de correction doit, de plus, être préservé par l'élimination des coupures pour garantir que les calculs effectués sur les réseaux ne détruisent pas le fait de correspondre à une preuve. Nous présentons ici le critère de correction le plus répandu : le critère de Danos-Regnier [DR89], mais il en existe de nombreux autres (voir par exemple [Gir87, Mét94, Lam94, Gue99]).

1.7. Définition (Structure de preuve séquentialisable)
On donne une définition récursive du fait que la structure de preuve $\mathcal{S}$ est *séquentialisable* :
- Si $\mathcal{S}$ est constituée d'un unique lien *ax*, $\mathcal{S}$ est séquentialisable.
- Si $\mathcal{S}$ possède un lien *cut* ou $\otimes$ terminal et si en le supprimant on obtient deux structures de preuve séquentialisables alors $\mathcal{S}$ est séquentialisable.
- Si $\mathcal{S}$ possède un lien ⅋, $?d$, $?c$, $?w$, $\forall$, ou $\exists$ terminal et si en le supprimant on obtient une structure de preuve séquentialisable alors $\mathcal{S}$ est séquentialisable.
- Si $\mathcal{S}$ possède, comme unique lien à profondeur 0, un lien ! et si la boîte associée à ce nœud est séquentialisable alors $\mathcal{S}$ est séquentialisable.

1.8. Lemme
*Une structure de preuve est séquentialisable* ssi *elle est la traduction d'une preuve du calcul des séquents.*

1.9. Définition (Graphes de correction)
Les *graphes de correction* à profondeur 0 d'une structure de preuve $\mathcal{S}$ sont les graphes (non orientés) formés des nœuds à profondeur 0 de $\mathcal{S}$ dans lesquels :

- On supprime les arêtes pendantes.
- Chaque lien ! dont la boîte a pour conclusions $?\Gamma, A$ est remplacé par un lien ! généralisé sans prémisse et avec conclusions $?\Gamma, !A$ :




- Pour chaque lien $\parr$, une des deux arêtes prémisses est supprimée.
- Pour chaque lien $?c$, une des deux arêtes prémisses est supprimée.
- Si $n$ est un lien $\forall$, son arête prémisse peut être remplacée par une arête entre $n$ et un lien qui dépend de sa variable propre.

Un graphe de correction de $\mathcal{S}$ est soit un graphe de correction à profondeur 0 de $\mathcal{S}$ soit un graphe de correction à profondeur 0 d'une des boîtes de $\mathcal{S}$.

1.10. Définition (Réseau de preuve)
Une structure de preuve sans affaiblissement est un *réseau de preuve* ou est *correcte* si tous ses graphes de correction sont acycliques et connexes.

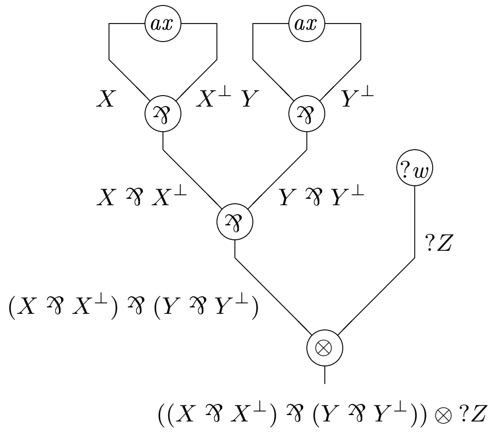
1.11. Théorème (Séquentialisation)
*Une structure de preuve sans affaiblissement est séquentialisable* ssi *elle est correcte.*

Trouver un critère de correction pour ces structures de preuve avec affaiblissement est actuellement un problème ouvert. La difficulté vient du fait que l'affaiblissement introduit des disconnexités dans les graphes de correction. Une possibilité serait de remplacer la contrainte de connexité par une comptabilité du nombre de composantes connexes, en demandant :

(C) Les graphes de correction sont acycliques et ont une composante connexe de plus que de liens $?w$.

Toutefois cette condition est nécessaire pour la séquentialisation mais pas suffisante comme le montre l'exemple 1.12.

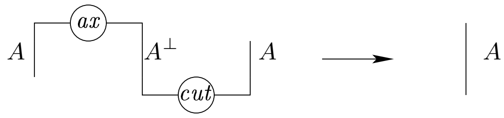
1.12. Exemple
La formule $((X \parr X^\perp) \parr (Y \parr Y^\perp)) \otimes ?Z$ n'est pas prouvable dans $\mathsf{LL}$ (car, en particulier, $?Z$ ne l'est pas) mais la structure de preuve suivante a ses 8 graphes de correction acycliques et qui contiennent 2 composantes connexes :



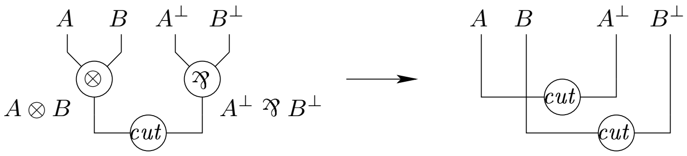
## 1.3 Élimination des coupures

1.13. Définition (Étapes d'élimination des coupures)
Pour chaque nœud coupure, on définit selon ses nœuds prémisses, les différentes étapes d'*élimination des coupures* :

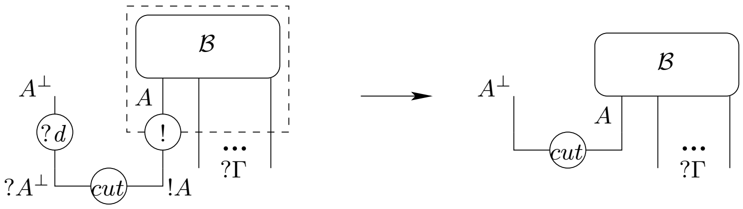
($ax$)



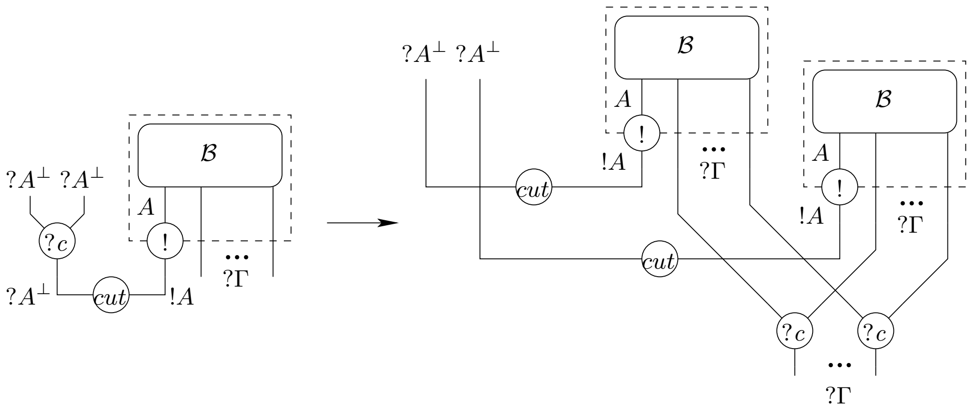

($\otimes$/ $\parr$)



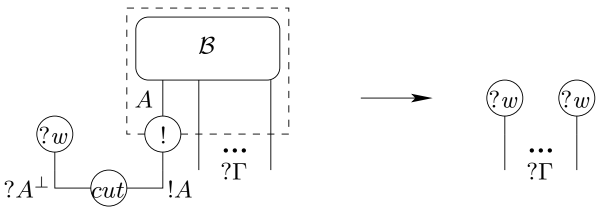

($!/?\,d$)



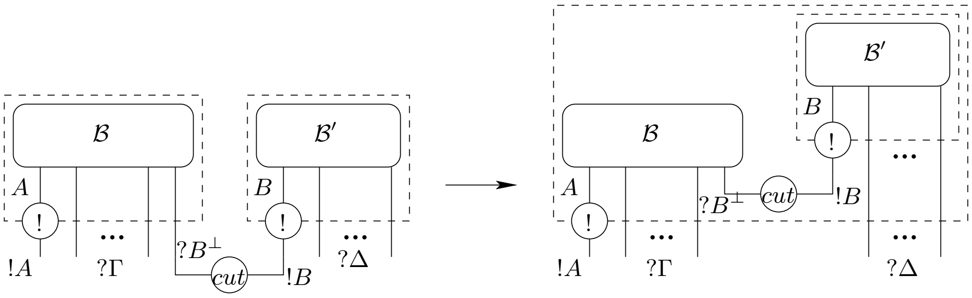

($!/?\,c$)



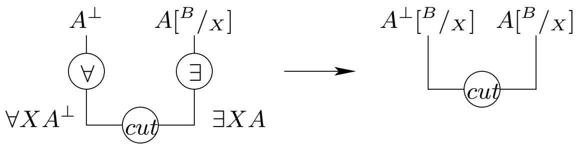

($!/?\,w$)



(!/!)




(∀/∃)




$X$ est substituée par $B$ dans toute la structure de preuve.

1.14. Proposition (Préservation de la correction)
*Soit $\mathcal{R}$ un réseau de preuve et $\mathcal{R}'$ un réduit de $\mathcal{R}$, $\mathcal{R}'$ est un réseau de preuve.*

1.15. Remarque : si $\mathcal{R}$ contient des affaiblissements, on ne prend pas en compte la connexité mais la condition (C) est préservée.

1.16. Théorème (Normalisation forte et Confluence)
*La réduction des réseaux de preuve est* fortement normalisante *et* confluente.

# Chapitre 2

# Sémantiques de LL

Nous rappelons dans ce chapitre deux sémantiques dénotationnelles caractéristiques de LL. Tout d'abord la sémantique cohérente [Gir87] qui a donné naissance à LL en décomposant l'implication intuitionniste. Ensuite nous aborderons la géométrie de l'interaction [Gir88] pour MELL, cette interprétation géométrique des preuves de logique linéaire est en particulier à la base de la sémantique des jeux et de l'analyse logique de la réduction optimale du $\lambda$-calcul.

## 2.1 Sémantique cohérente

Nous reprenons ici les définitions de [Gir95b] concernant les espaces cohérents. Il s'agit de ce que l'on appelle la sémantique cohérente *multi-ensembliste* alors que les définitions originelles de [Gir87] étaient basées sur une utilisation d'ensembles pour interpréter les connecteurs exponentiels.

NOTATIONS :

ENSEMBLES.

- On note $a \subset b$ l'inclusion (au sens large) de deux ensembles et $a \subsetneq b$ l'inclusion stricte.
- La *somme disjointe* $a + b$ de deux ensembles $a$ et $b$ est l'ensemble $(\{1\} \times a) \cup (\{2\} \times b)$.

MULTI-ENSEMBLES.

- Soit $a$ un ensemble, un *multi-ensemble* $\mu$ sur $a$ est une fonction de $a$ dans $\mathbb{N}$.
  Le support $|\mu|$ du multi-ensemble $\mu$ est le sous-ensemble de $a$ constitué des éléments dont l'image par $\mu$ est non nulle.
  Un multi-ensemble est dit *fini* si son support est fini.
- Le multi-ensemble vide (dont le support est vide, c'est-à-dire le multi-ensemble correspondant à la fonction nulle) est noté $\varepsilon$ ou $[\,]$.
- La somme (ou union) de deux multi-ensembles est tout simplement la somme des fonctions, avec comme élément neutre $\varepsilon$.
- Le multi-ensemble fini $\mu$ sera noté entre crochets avec chaque élément $x \in |\mu|$ répété $\mu(x)$ fois.

2.1. EXEMPLE
Soient $\mu$ le multi-ensemble sur $\{x, y, z\}$ suivant :

$$\begin{aligned} x &\mapsto 2 \\ y &\mapsto 3 \\ z &\mapsto 0 \end{aligned}$$

et $\nu$ le multi-ensemble sur $\{y, z, t\}$ :

$$\begin{aligned} y &\mapsto 2 \\ z &\mapsto 1 \\ t &\mapsto 1 \end{aligned}$$

$\mu$ et $\nu$ sont des multi-ensembles finis et on les note $\mu = [x, x, y, y, y]$ et $\nu = [y, y, z, t]$. Le support de $\mu$ est $|\mu| = \{x, y\}$ et celui de $\nu$ est $|\nu| = \{y, z, t\}$. La somme $\mu+\nu$ est notée $\mu+\nu = [x, x, y, y, y, y, y, z, t]$ et a pour support $|\mu + \nu| = |\mu| \cup |\nu| = \{x, y, z, t\}$.

2.2. Définition (Espace cohérent)
Un *espace cohérent* $\mathcal{A}$ est un couple $(|\mathcal{A}|, \mathrel{\overset{\frown}{\smile}}_{\mathcal{A}})$ où $|\mathcal{A}|$ est un ensemble (la *trame* de $\mathcal{A}$) et $\mathrel{\overset{\frown}{\smile}}_{\mathcal{A}}$ est une relation symétrique et réflexive sur $|\mathcal{A}|$.

Une *clique* $a$ d'un espace cohérent $\mathcal{A}$, noté $a \sqsubset \mathcal{A}$, est un sous-ensemble de $|\mathcal{A}|$ tel que quels que soient $x$ et $y$ dans $a$, on ait $x \mathrel{\overset{\frown}{\smile}}_{\mathcal{A}} y$.

Notations : Lorsqu'aucune confusion n'est possible, on abrégera $\mathrel{\overset{\frown}{\smile}}_{\mathcal{A}}$ en $\mathrel{\overset{\frown}{\smile}}$.

On définit les autres relations de cohérence à partir de $\mathrel{\overset{\frown}{\smile}}$ :
- Cohérence stricte : $x \frown y$ si $x \mathrel{\overset{\frown}{\smile}} y$ et $x \neq y$.
- Incohérence : $x \asymp y$ est le contraire de $x \frown y$.
- Incohérence stricte : $x \smile y$ si $x \asymp y$ et $x \neq y$ (c'est donc le contraire de $x \mathrel{\overset{\frown}{\smile}} y$).

2.3. Remarque : La donnée d'une seule des relations $\mathrel{\overset{\frown}{\smile}}$, $\frown$, $\asymp$ et $\smile$ suffit à déterminer toutes les autres.

2.4. Définition (Constructions)
Soient $\mathcal{A}$ et $\mathcal{B}$ deux espaces cohérents, on définit les espaces cohérents suivants :

$\mathcal{A}^\perp$ a pour trame $|\mathcal{A}|$ et $x \mathrel{\overset{\frown}{\smile}}_{\mathcal{A}^\perp} x'$ si $x \asymp_{\mathcal{A}} x'$.

$\mathcal{A} \otimes \mathcal{B}$ a pour trame $|\mathcal{A}| \times |\mathcal{B}|$ et $(x, y) \mathrel{\overset{\frown}{\smile}}_{\mathcal{A}\otimes\mathcal{B}} (x', y')$ si $x \mathrel{\overset{\frown}{\smile}}_{\mathcal{A}} x'$ et $y \mathrel{\overset{\frown}{\smile}}_{\mathcal{B}} y'$.

$\mathcal{A} \parr \mathcal{B}$ a pour trame $|\mathcal{A}| \times |\mathcal{B}|$ et $(x, y) \frown_{\mathcal{A}\parr\mathcal{B}} (x', y')$ si $x \frown_{\mathcal{A}} x'$ ou $y \frown_{\mathcal{B}} y'$.

$\mathcal{A} \oplus \mathcal{B}$ a pour trame $|\mathcal{A}| + |\mathcal{B}|$ (somme disjointe ensembliste) et $(i, x) \mathrel{\overset{\frown}{\smile}}_{\mathcal{A}\oplus\mathcal{B}} (i, x')$ si $x \mathrel{\overset{\frown}{\smile}}_{\mathcal{A}} x'$ et $(i, x) \smile_{\mathcal{A}\oplus\mathcal{B}} (j, x')$ si $i \neq j$ quels que soient $x$ et $y$ avec $i, j \in \{1, 2\}$.

$\mathcal{A} \,\&\, \mathcal{B}$ a pour trame $|\mathcal{A}| + |\mathcal{B}|$ et $(i, x) \mathrel{\overset{\frown}{\smile}}_{\mathcal{A}\&\mathcal{B}} (i, x')$ si $x \mathrel{\overset{\frown}{\smile}}_{\mathcal{A}} x'$ et $(i, x) \frown_{\mathcal{A}\&\mathcal{B}} (j, x')$ si $i \neq j$ quels que soient $x$ et $y$ avec $i, j \in \{1, 2\}$.

$1$ a pour trame $\{\star\}$ en remarquant qu'à isomorphisme près il n'y a qu'un espace cohérent à un point.

$\perp$ est également $\{\star\}$.

$0$ a pour trame $\emptyset$, l'espace cohérent à trame vide.

$\top$ est également vide.

$!\mathcal{A}$ a pour trame l'ensemble des multi-ensembles finis d'éléments de $|\mathcal{A}|$ dont les supports sont des cliques de $\mathcal{A}$ et $x \mathrel{\overset{\frown}{\smile}}_{!\mathcal{A}} y$ si $|x + y| \sqsubset \mathcal{A}$.

$?\mathcal{A}$ a pour trame l'ensemble des multi-ensembles finis d'éléments de $|\mathcal{A}|$ dont les supports sont des *anti-cliques* de $\mathcal{A}$ (*i.e.* des cliques de $\mathcal{A}^\perp$) et $x \frown_{?\mathcal{A}} y$ si $|x+y|$ n'est pas une anti-clique de $\mathcal{A}$.

2.5. Exemple (Cliques)
Soit $\mathcal{A}$ un espace cohérent quelconque, l'ensemble vide $\emptyset$ forme une clique de $\mathcal{A}$, appelée clique vide.

Si $x$ et $y$ sont deux éléments de $|\mathcal{A}|$, $(x, x)$ et $(y, y)$ sont cohérents dans $\mathcal{A} \parr \mathcal{A}^\perp$ (car soit $x \frown_{\mathcal{A}} y$, soit $x \frown_{\mathcal{A}^\perp} y$, soit $x = y$), donc $\{(x, x) \mid x \in |\mathcal{A}|\} \sqsubset \mathcal{A} \parr \mathcal{A}^\perp$.

Les espaces cohérents permettent de définir un modèle de $\mathsf{LL}$. Nous rappelons le cas propositionnel :

2.6. DÉFINITION (INTERPRÉTATION DE $\mathsf{LL}$)
On interprète les atomes par des espaces cohérents quelconques, les formules de $\mathsf{LL}$ par l'espace cohérent correspondant. Une preuve $\pi$ du séquent $\vdash A_1, ..., A_n$ est interprétée par une clique $[\![\pi]\!]_{coh}$ de l'espace cohérent $\mathcal{A}_1 \parr ... \parr \mathcal{A}_n$ selon la dernière règle de $\pi$. On note $\pi_1$ la prémisse de cette dernière règle si elle est unaire ou $\pi_1$ et $\pi_2$ si elle est binaire.

$(ax)$ $[\![\pi]\!]_{coh} = \{(x, x) \mid x \in |\mathcal{A}|\}$

$(cut)$ $[\![\pi]\!]_{coh} = \{(x_1, ..., x_n, x'_1, ..., x'_m) \mid \exists y,\ (x_1, ..., x_n, y) \in [\![\pi_1]\!]_{coh} \wedge (y, x'_1, ..., x'_m) \in [\![\pi_2]\!]_{coh}\}$

$(\parr)$ $[\![\pi]\!]_{coh} = \{(x_1, ..., x_n, (y, z)) \mid (x_1, ..., x_n, y, z) \in [\![\pi_1]\!]_{coh}\}$

$(\otimes)$ $[\![\pi]\!]_{coh} = \{(x_1, ..., x_n, x'_1, ..., x'_m, (y, y')) \mid (x_1, ..., x_n, y) \in [\![\pi_1]\!]_{coh} \wedge (x'_1, ..., x'_m, y') \in [\![\pi_2]\!]_{coh}\}$

$(\&)$ $[\![\pi]\!]_{coh} = \{(x_1, ..., x_n, (1, y)) \mid (x_1, ..., x_n, y) \in [\![\pi_1]\!]_{coh}\} \cup \{(x_1, ..., x_n, (2, y)) \mid (x_1, ..., x_n, y) \in [\![\pi_2]\!]_{coh}\}$

$(\oplus_i)$ $[\![\pi]\!]_{coh} = \{(x_1, ..., x_n, (i, y)) \mid (x_1, ..., x_n, y) \in [\![\pi_1]\!]_{coh}\}$

$(!)$ $[\![\pi]\!]_{coh} = \{(\sum_{i=1}^{k} x_1^i, ..., \sum_{i=1}^{k} x_n^i, [y^1, ... y^k]) \mid \forall 1 \leq i \leq k,\ (x_1^i, ..., x_n^i, y^i) \in [\![\pi_1]\!]_{coh} \wedge \forall 1 \leq i' \leq k,\ \forall 1 \leq j \leq n,\ x_j^i \frown x_j^{i'}\}$

$(?d)$ $[\![\pi]\!]_{coh} = \{(x_1, ..., x_n, [y]) \mid (x_1, ..., x_n, y) \in [\![\pi_1]\!]_{coh}\}$

$(?c)$ $[\![\pi]\!]_{coh} = \{(x_1, ..., x_n, y + z) \mid (x_1, ..., x_n, y, z) \in [\![\pi_1]\!]_{coh} \wedge y \asymp z\}$

$(?w)$ $[\![\pi]\!]_{coh} = \{(x_1, ..., x_n, \varepsilon) \mid (x_1, ..., x_n) \in [\![\pi_1]\!]_{coh}\}$

$(\perp)$ $[\![\pi]\!]_{coh} = \{(x_1, ..., x_n, \star) \mid (x_1, ..., x_n) \in [\![\pi_1]\!]_{coh}\}$

$(1)$ $[\![\pi]\!]_{coh} = \{\star\}$

$(\top)$ $[\![\pi]\!]_{coh} = \emptyset$

2.7. THÉORÈME (MODÈLE COHÉRENT)
*Les espaces cohérents sont un modèle dénotationnel de* $\mathsf{LL}$*,* i.e. *si $\pi$ est une preuve qui se réduit en $\pi'$ par élimination des coupures alors* $[\![\pi]\!]_{coh} = [\![\pi']\!]_{coh}$.

## 2.2 Géométrie de l'interaction

Les principaux résultats de la GoI pour $\mathsf{MELL}$ que nous donnons ici ont été initialement décrits dans [Gir88, DR95, DR99, DHR96]. De nombreuses présentations de la GoI ont été données : $C^\star$-algèbres, semi-goupes inversifs, algèbres de clauses, automates bi-déterministes, ... Une idée commune à toutes ces présentations est celle de *chemin*. Par chemin dans une preuve ou un réseau, nous entendrons chemin dans les formules partant d'une conclusion, montant jusqu'à un axiome, redescendant par l'autre formule jusqu'à une coupure, remontant jusqu'à un axiome, ... et terminant dans une conclusion. On interprète une preuve par l'ensemble de ses chemins *persistants*, *i.e.* qui ne sont pas coupés par élimination des coupures. L'objectif de la GoI est de donner une caractérisation algébrique de ces chemins. Concrètement, on définit algébriquement une famille de chemins que l'on appelle *chemins réguliers* et on montre qu'un chemin est régulier si et seulement si il est persistant.

Nous utiliserons ici la présentation qui consiste à interpréter les preuves (ou les réseaux) comme des machines à jetons, les chemins réguliers sont alors ceux tels qu'il existe jeton qui parcourt ce chemin lors d'une exécution de la machine.

2.8. Définition (Jeton)
Un *jeton* est un triplet $(m, b, s)$ où :

- $m$ est une pile construite à partir des symboles $g$ et $d$ ($\varepsilon$ est la pile vide et $|.|$ la longueur d'une pile), appelée *jeton multiplicatif*,
- $b$ et $s$ sont des piles de signatures exponentielles ou *piles exponentielles*, appelées respectivement *pile de boîtes* et *pile d'équilibrage*,
- une *signature exponentielle* $\sigma$ est définie par la grammaire :

$$\sigma \quad ::= \quad \star \quad | \quad g.\sigma \quad | \quad d.\sigma \quad | \quad \ulcorner\sigma\urcorner.\sigma$$

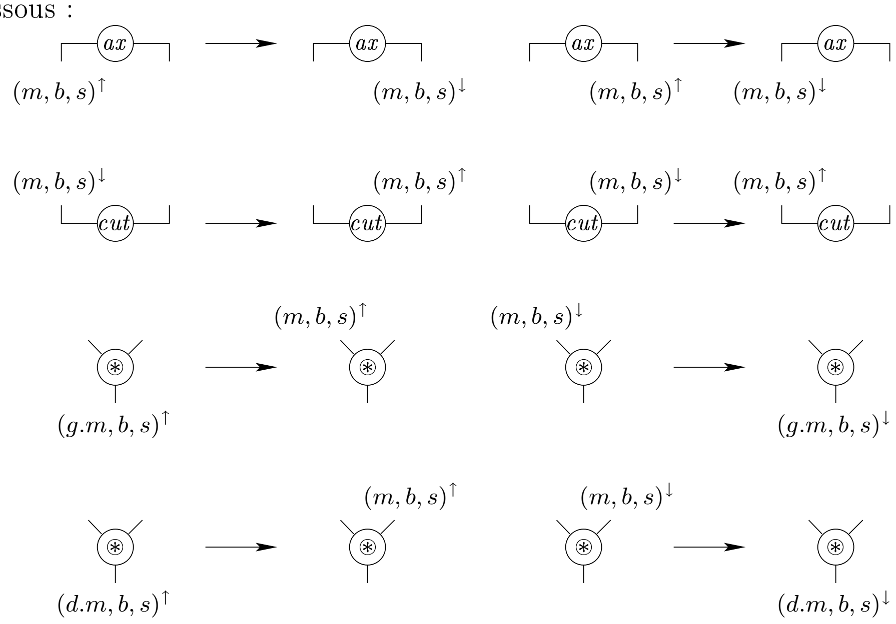
2.9. Définition (Profondeur exponentielle d'un jeton multiplicatif)
Soit $m$ un jeton multiplicatif et $A$ une formule de $\mathsf{MELL}$, la *profondeur exponentielle* de $m$ dans $A$, notée $\mathit{prof}_A(m)$, est définie par :

- $\mathit{prof}_X(\varepsilon) = \mathit{prof}_{B\otimes C}(\varepsilon) = \mathit{prof}_{B\parr C}(\varepsilon) = 0$,
- $\mathit{prof}_X(m)$ n'est pas définie si $m \neq \varepsilon$,
- $\mathit{prof}_{B\otimes C}(g.m) = \mathit{prof}_{B\parr C}(g.m) = \mathit{prof}_B(m)$,
- $\mathit{prof}_{B\otimes C}(d.m) = \mathit{prof}_{B\parr C}(d.m) = \mathit{prof}_C(m)$,
- $\mathit{prof}_{?B}(m) = \mathit{prof}_{!B}(m) = \mathit{prof}_B(m) + 1$.

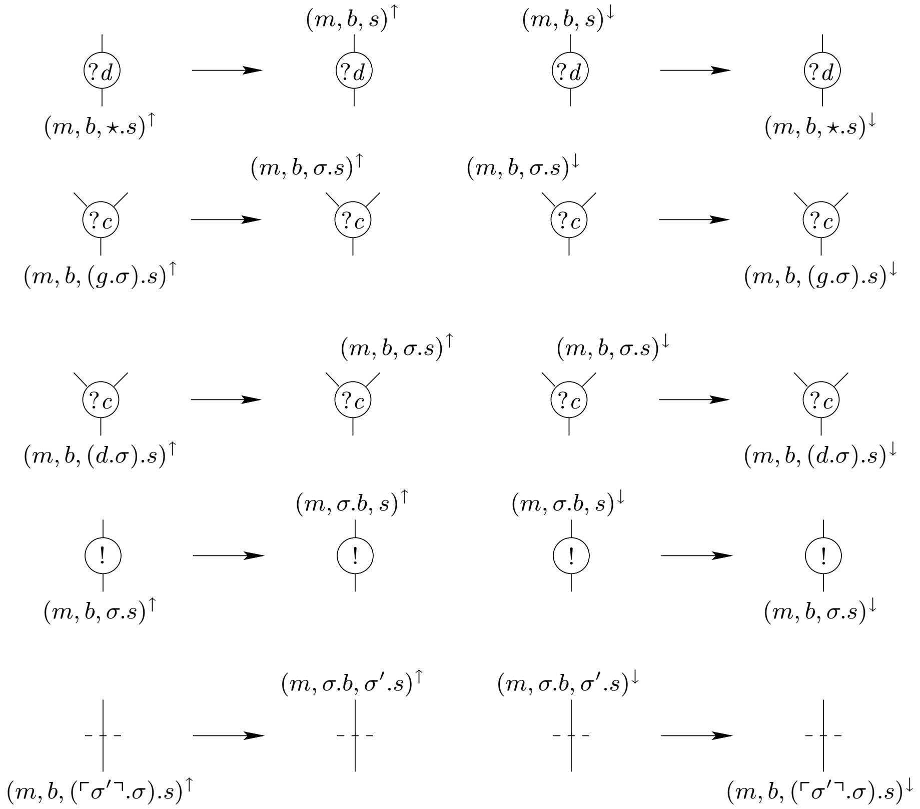
2.10. Remarque : Intuitivement, la profondeur exponentielle d'un jeton multiplicatif $m$ dans une formule $A$ est le nombre de connecteurs exponentiels dans le champ desquels la sous-formule multiplicative de $A$ décrite par $m$ se trouve.

2.11. Définition (Validité d'un jeton)
Soit $\mathcal{R}$ un réseau et $a$ une arête de type $A$ de $\mathcal{R}$, le jeton $(m, b, s)$ est *valide* pour $a$ si $|b|$ est la profondeur de $a$ dans $\mathcal{R}$, $\mathit{prof}_A(m)$ est bien définie et $|s| = \mathit{prof}_A(m)$.

2.12. Définition (Machine abstraite d'interaction)
Soit $\mathcal{R}$ un réseau de $\mathsf{MELL}$, on définit la *machine abstraite d'interaction* (ou IAM) $M_{\mathcal{R}}$ associée à $\mathcal{R}$. Un état de $M_{\mathcal{R}}$ est la donnée d'une arête $a$ dans $\mathcal{R}$, d'une direction $\updownarrow$ qui est soit $\uparrow$ soit $\downarrow$ et d'un jeton $(m, b, s)$ valide pour $a$, ou bien l'état particulier $\emptyset$ qui signifie que la machine s'arrête (l'intuition étant qu'il s'agit d'un message d'erreur). Une transition de $M_{\mathcal{R}}$ à partir de l'état $A^{\updownarrow}(m, b, s)$ est donnée par le nœud situé au-dessus de $A$ si $\updownarrow=\uparrow$ et par le nœud situé au-dessous de $A$ si $\updownarrow=\downarrow$, comme décrit ci-dessous :



où le nœud $\circledast$ est soit un nœud $\otimes$ soit un nœud $\parr$.




Lorsqu'un jeton franchit une porte auxiliaire de boîte, il est modifié comme s'il avait traversé un nœud. Aucune transition n'est définie pour les nœuds $?w$.

Un jeton qui monte dans un nœud $?d$, $?c$ ou une porte auxiliaire peut avoir une jeton qui ne correspond pas à l'une des transitions décrites ci-dessus. Dans ce cas, la machine passe dans l'état $\emptyset$.

Un état initial (resp. final) de la machine est un état dont la formule est une conclusion de $\mathcal{R}$ et dont la direction est $\uparrow$ (resp. $\downarrow$).

La *fonction d'interaction* $f_{\mathcal{R}}$ est la fonction partielle qui, à tout état initial, associe l'état final (ou $\emptyset$) atteint par $M_{\mathcal{R}}$ à partir de cet état initial, s'il existe.

2.13. Théorème (Correction)
*Soit $\mathcal{R}$ un réseau sans ? dans ses conclusions, si $\mathcal{R}$ se réduit en une forme normale $\mathcal{R}_0$ alors $f_{\mathcal{R}} = f_{\mathcal{R}_0}$.*

Démonstration : La preuve consiste à montrer que $f_{\mathcal{R}}$ est préservée à travers les étapes de réduction qui, dans le cas des coupures exponentielles, ne concernent que des boîtes sans contexte. Ces étapes étant suffisantes pour atteindre une forme normale à partir d'un réseau sans conclusion $?A$. On conclut alors par confluence de la réduction de MELL. □

2.14. Corollaire (Terminaison)
*Soit $\mathcal{R}$ un réseau, la fonction $f_{\mathcal{R}}$ est totale,* i.e. *le calcul de la machine $M_{\mathcal{R}}$ à partir d'un état initial termine.*

2.15. Remarque : La contrainte imposant qu'il n'y ait pas de ? dans les conclusions des preuves est forte mais permet néanmoins de calculer. Par exemple, il est possible de coder les booléens à l'aide des preuves de $\vdash X^\perp \parr X^\perp, X \otimes X$ (puisqu'il en existe exactement deux distinctes). Un calcul dont le résultat final est un booléen correspond ainsi à la réduction d'un réseau de conclusions $X^\perp \parr X^\perp$ et $X \otimes X$ qui ne contiennent pas de ? et donc pour lequel l'interprétation par la GoI est correcte.

Il est possible de raffiner ce résultat de correction en autorisant les ? mais en ne considérant que les étapes de réduction $!/?w$ lorsque la boîte n'a pas de portes auxiliaires. Dans ce cas on n'a pas exactement $f_{\mathcal{R}} = f_{\mathcal{R}_0}$ mais on peut montrer que les chemins parcourus par les jetons dans $\mathcal{R}$ et $\mathcal{R}_0$ se correspondent (comme décrit dans [DR95]). Pour cela, une méthode consiste à considérer une variante de MELL, que l'on appellera MELL avec *promotion fonctorielle*, où la règle ! est remplacée par les deux règles :

$$\frac{\vdash \Gamma, A}{\vdash ?\Gamma, !A}\,! \qquad \frac{\vdash \Gamma, ??A}{\vdash \Gamma, ?A}\,??$$

il est clair que la règle usuelle est codable avec ces deux règles et que, réciproquement, ces règles sont codables avec les règles usuelles. Le choix de ces règles est discuté un peu plus longuement dans [Gir95a].

On modifie alors légèrement l'élimination des coupures par :

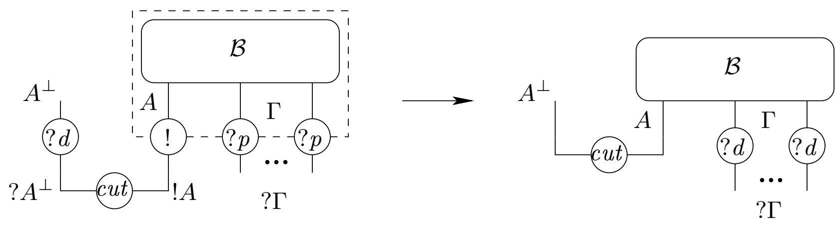
$$\frac{\dfrac{\vdash \Gamma, A}{\vdash ?\Gamma, !A}\,! \qquad \dfrac{\vdash \Delta, A^\perp}{\vdash \Delta, ?A^\perp}\,?d}{\vdash ?\Gamma, \Delta}\,cut \quad\longrightarrow\quad \dfrac{\dfrac{\vdash \Gamma, A \qquad \vdash \Delta, A^\perp}{\vdash \Gamma, \Delta}\,cut}{\vdash ?\Gamma, \Delta}\,?d$$

$$\frac{\dfrac{\vdash \Gamma, A}{\vdash ?\Gamma, !A}\,! \qquad \dfrac{\vdash \Delta, ??A^\perp}{\vdash \Delta, ?A^\perp}\,??}{\vdash ?\Gamma, \Delta}\,cut \quad\longrightarrow\quad \dfrac{\dfrac{\dfrac{\dfrac{\vdash \Gamma, A}{\vdash ?\Gamma, !A}\,!}{\vdash ??\Gamma, !!A}\,! \qquad \vdash \Delta, ??A^\perp}{\vdash ??\Gamma, \Delta}\,cut}{\vdash ?\Gamma, \Delta}\,??$$

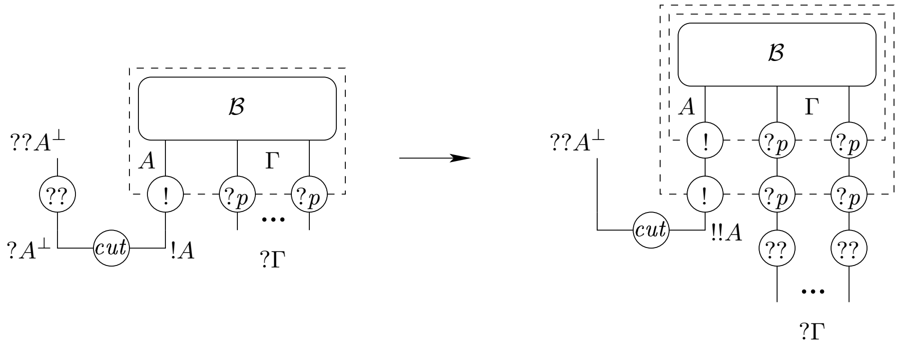
$$\frac{\dfrac{\vdash \Gamma, B^\perp, A}{\vdash ?\Gamma, ?B^\perp, !A}\,! \qquad \dfrac{\vdash \Delta, B}{\vdash ?\Delta, !B}\,!}{\vdash ?\Gamma, ?\Delta, !A}\,cut \quad\longrightarrow\quad \dfrac{\dfrac{\vdash \Gamma, B^\perp, A \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A}\,cut}{\vdash ?\Gamma, ?\Delta, !A}\,!$$

Dans les réseaux, pour interpréter la nouvelle règle !, on introduit un nouveau nœud $?p$ unaire dont la prémisse a type $A$ et la conclusion a type $?A$. À chaque nœud ! est associée une boîte dont les conclusions sont prémisses de nœuds $?p$ (on dit que ces nœuds $?p$ sont associés au nœud !). Tout nœud $?p$ doit être associé à un nœud !. La nouvelle règle ?? est interprétée par un nouveau nœud unaire. Ce qui donne pour l'élimination des coupures :

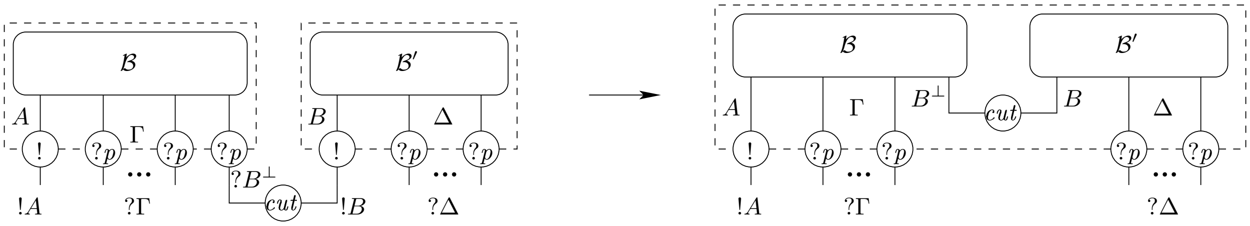
$(!/?d)$



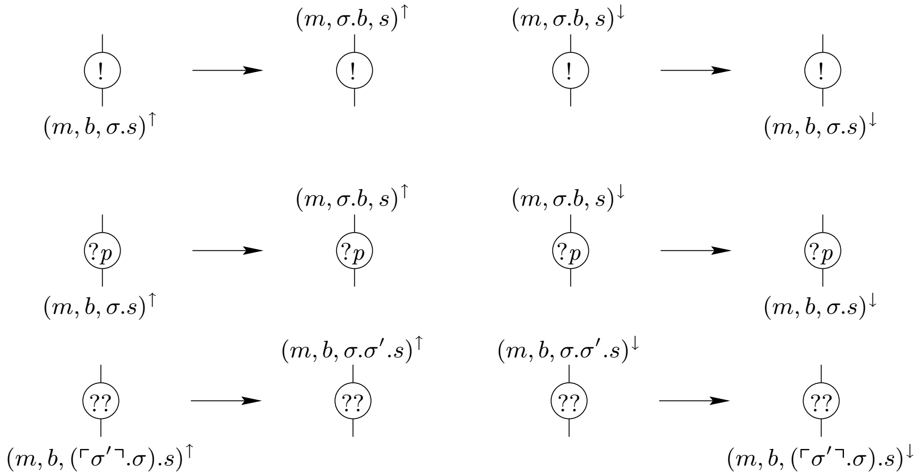
(!/??)




(!/!)




Les transitions de la machine à jetons sont :




qui sont compatibles avec le codage des anciennes règles dans les nouvelles.

# Chapitre 3

# Sémantique des jeux

La sémantique des jeux a été utilisée pour fournir à la fois des modèles de systèmes logiques et de langages de programmation. Nous nous intéressons ici à ce qui concerne la logique linéaire intuitionniste et la logique intuitionniste.

Les définitions des jeux Abramsky-Jagadeesan-Malacaria (ou $AJM$) et celles des jeux Hyland-Ong/Nickau (ou $HO$) sont essentiellement reprises de [AJM00, HO00, McC96] même si quelques nouveautés sont introduites dans la présentation (en particulier le fait de mettre en avant les notions de $a$-morphisme et $p$-morphisme).

Pour une introduction plus détaillée aux jeux, on pourra se référer aux deux thèses [McC96] et [Har99].

## 3.1 Jeux linéaires HO

3.1. Définition (Arène HO)
Une *arène* $A$ est un triplet $(M_A, \lambda_A, \vdash_A)$ où :

- $M_A$ est un ensemble appelé ensemble de *coups*,
- $\lambda_A$ est une fonction de $M_A$ dans $\{O, J\}$,
- $\vdash_A$ est la donnée d'un sous-ensemble $M_A^i$ de $M_A$ appelé ensemble des *coups initiaux* (on note $\vdash_A m$ si $m$ est dans cet ensemble et on note $M_A^{ni}$ l'ensemble des coups non initiaux) et d'une relation binaire sur $M_A$ appelée *relation de validation* (on note $m \vdash_A n$ si $m$ *valide* $n$). $\vdash_A$ doit vérifier :
  - Si $\vdash_A m$ alors $\lambda_A(m) = O$ et pour tout coup $n \in M_A$, $n \nvdash_A m$.
  - Si $m \vdash_A n$ alors $\lambda_A(m) \neq \lambda_A(n)$

3.2. Définition (Suite justifiée)
Une *suite justifiée* de l'arène $A$ est une paire $(s, f)$ où $s = m_1 ... m_n$ est une suite de coups de $A$ et $f$ est une fonction de $\{1, ..., n\}$ dans $\{0, ..., n-1\}$ telle que pour tout $1 \leq i \leq n$ :

- $f(i) < i$
- si $f(i) = 0$ alors $\vdash_A m_i$
- si $f(i) = j \neq 0$ alors $m_j \vdash_A m_i$

Lorsque $f(i) = j \neq 0$, on dit que le coup $m_i$ *pointe* sur $m_j$ ou que $m_i$ est *justifié* par $m_j$. Et lorsqu'il existe un $k \geq 0$ tel que $f^k(i) = j \neq 0$, on dit que $m_i$ est *héréditairement justifié* par $m_j$.

Une suite justifiée est *bien ouverte* si $f(i) = 0$ uniquement pour $i = 1$, *i.e.* si son seul coup initial est son premier coup.

Cette définition rigoureuse étant donnée, on ne fera presque plus jamais référence à la fonction $f$. On considérera qu'à chaque coup d'une suite justifiée est associé un coup situé avant dans la suite (ou rien dans le cas des coups initiaux), le défaut de ce point de vue provient bien entendu du fait qu'un coup peut apparaître deux fois dans une suite : dans $abac$ si on dit que $c$ est justifié par $a$, il y a une ambiguïté. Dans la pratique ces ambiguïtés seront toujours facilement évitées.

NOTATIONS (SUITES) : Si $t$ est une suite justifiée terminant par un coup $m$ tel que $\lambda_A(m) = J$ (resp. $\lambda_A(m) = O$), et si $s$ est la sous-suite de $t$ ne contenant pas ce dernier coup $m$, on note $t = sm^J$ (resp. $t = sm^O$).

On note $s \leq t$ l'ordre préfixe sur les suites et $\varepsilon$ la suite vide.

Si $s$ est une suite justifiée, on note $s \leq^J t$ si $s \leq t$ et le dernier élément de $s$ est un coup $J$ (resp. $O$), on dit que $s$ est un *$J$-préfixe* (resp. *$O$-préfixe*) de $t$.

3.3. DÉFINITION (PROJECTION)
Soit $s$ une suite justifiée et $I$ un ensemble non vide de coups initiaux de $s$, on définit la *projection* $s \upharpoonright_I$ de $s$ sur $I$ comme la sous-suite de $s$ formée uniquement des coups héréditairement justifiés par un élément de $I$.

3.4. DÉFINITION (VUE)
Soit $s$ une suite justifiée, on définit la *vue joueur* de $s$, notée $\ulcorner s \urcorner$, qui est une sous-suite justifiée, par :
- $\ulcorner \varepsilon \urcorner = \varepsilon$
- $\ulcorner sm^J \urcorner = \ulcorner s \urcorner m^J$
- $\ulcorner sm^O \urcorner = m^O$ si $m$ est initial
- $\ulcorner smtn^O \urcorner = \ulcorner s \urcorner mn^O$ si $n$ est justifié par $m$

où $s$ et $t$ sont des suites.

La *vue opposant* est définie de manière similaire par :
- $\llcorner \varepsilon \lrcorner = \varepsilon$
- $\llcorner sm^O \lrcorner = \llcorner s \lrcorner m^O$
- $\llcorner smtn^J \lrcorner = \llcorner s \lrcorner mn^J$ si $n$ est justifié par $m$

3.5. DÉFINITION (POSITION LÉGALE)
Une *position légale* est une suite justifiée $s$ telle que :
- Si $tmn \leq s$ alors $\lambda(m) \neq \lambda(n)$.
- Si $tm^J \leq s$ alors $m$ pointe dans $\ulcorner t \urcorner$.
- Si $tm^O \leq s$ et $m$ n'est pas initial alors $m$ pointe dans $\llcorner t \lrcorner$.

On note $L_A$ l'ensemble des positions légales de l'arène $A$.

Puisque si $n$ est justifié par $m$ dans $s$ et $n$ est dans $\ulcorner s \urcorner$ alors $m$ est dans $\ulcorner s \urcorner$, on conserve dans $\ulcorner s \urcorner$ les mêmes pointeurs que dans $s$.

3.6. DÉFINITION (CONSTRUCTIONS D'ARÈNES)
SOMME. Soient $A$ et $B$ deux arènes, on note $A + B$ l'*arène somme* définie par :
- $M_{A+B} = M_A + M_B$ et on identifiera $(i, x)$ et $x$ quand cela ne porte pas à confusion.
- $\lambda_{A+B} = [\lambda_A, \lambda_B]$, *i.e.* $\lambda_{A+B}(1, x) = \lambda_A(x)$ et $\lambda_{A+B}(2, x) = \lambda_B(x)$
- $\vdash_{A+B} m$ si $\vdash_A m$ ou $\vdash_B m$
- $m \vdash_{A+B} n$ si $m$ et $n$ sont dans $A$ et $m \vdash_A n$ ou si $m$ et $n$ sont dans $B$ et $m \vdash_B n$

Si $s$ est une position légale de $A + B$, on note $s \upharpoonright_A$ la sous-suite de $s$ ne contenant que les coups dans $A$, c'est une position légale de $A$ (et de même pour $B$).

FLÈCHE. Soient $A$ et $B$ deux arènes, on note $A \Rightarrow B$ l'*arène flèche* définie par :

- $M_{A\Rightarrow B} = M_A + M_B$
- $\lambda_{A\Rightarrow B} = [\overline{\lambda_A}, \lambda_B]$, *i.e.* $\lambda_{A\Rightarrow B}(1,x) \neq \lambda_A(x)$ et $\lambda_{A\Rightarrow B}(2,x) = \lambda_B(x)$
- $\vdash_{A\Rightarrow B} m$ si $\vdash_B m$
- $m \vdash_{A\Rightarrow B} n$ si $m$ et $n$ sont dans $A$ et $m \vdash_A n$, ou si $m$ et $n$ sont dans $B$ et $m \vdash_B n$, ou si $m$ est dans $B$, $n$ est dans $A$ et $\vdash_B m$ et $\vdash_A n$

Si $s$ est une position légale de $A \Rightarrow B$, on note $s \upharpoonright_A$ la sous-suite de $s$ ne contenant que les coups dans $A$, c'est une position légale de $A$ (et de même pour $B$).

3.7. Remarques :
- Soit $smn$ une position légale de l'arène $A+B$ avec $m \in M_A$ (resp. $m \in M_B$) et $n \in M_B$ (resp. $n \in M_A$), on a $\lambda_{A+B}(m) = J$ et $\lambda_{A+B}(n) = O$.
- Soit $smn$ une position légale de l'arène $A \Rightarrow B$ avec $m \in M_A$ (resp. $m \in M_B$) et $n \in M_B$ (resp. $n \in M_A$), on a $\lambda_{A\Rightarrow B}(m) = O$ et $\lambda_{A\Rightarrow B}(n) = J$.

C'est ce qu'on appelle habituellement les *conditions de switching*.

3.8. Définition (Jeu HO)
Un *jeu* $A$ est un quadruplet $(M_A, \lambda_A, \vdash_A, \mathcal{P}_A)$ où $(M_A, \lambda_A, \vdash_A)$ est une arène et $\mathcal{P}_A$, appelé l'ensemble des *parties* de $A$, est un ensemble de positions légales de cette arène non vide et clos par préfixe qui vérifie que :

(i) Si $I$ est un ensemble de coups initiaux de $s \in \mathcal{P}_A$ alors $s \upharpoonright_I \in \mathcal{P}_A$.

Un jeu est *bien ouvert* si toutes ses parties sont bien ouvertes.

Notation : $\mathcal{P}_A^J$ désigne l'ensemble de parties de $A$ qui terminent par un coup $J$.

3.9. Remarque : Lorsqu'un jeu est bien ouvert, la propriété (i) est trivialement vérifiée puisque $s \upharpoonright_I = s$.

3.10. Définition (Constructions linéaires de jeux)

($\top$) On note $\top$ l'unique jeu dont l'ensemble de coups est vide : $(\emptyset, \emptyset, \emptyset, \{\varepsilon\})$.

($\odot$) Si $A$ et $B$ sont deux jeux, le jeu $A \odot B$ a comme arène $A + B$ et comme ensemble de parties $\mathcal{P}_{A\odot B} = \{s \in L_{A+B} \mid s \upharpoonright_A \in \mathcal{P}_A \wedge s \upharpoonright_B \in \mathcal{P}_B\}$.

($\multimap$) Si $A$ et $B$ sont deux jeux, le jeu $A \multimap B$ a comme arène $A \Rightarrow B$ et comme ensemble de parties $\mathcal{P}_{A\multimap B} = \{s \in L_{A\Rightarrow B} \mid s \upharpoonright_A \in \mathcal{P}_A \wedge s \upharpoonright_B \in \mathcal{P}_B\}$.

(&) Si $A$ et $B$ sont deux jeux, le jeu $A \,\&\, B$ a comme arène $A + B$ et comme ensemble de parties $\mathcal{P}_{A\&B} = \mathcal{P}_A \cup \mathcal{P}_B$, la seule partie commune étant $\varepsilon$.

3.11. Définition (Morphismes)
Soient $A$ et $B$ deux jeux, on définit deux notions de morphismes entre jeux :
- Un *p-morphisme* (ou morphisme de parties) de $A$ dans $B$ est une fonction injective $f$ de $\mathcal{P}_A$ dans $\mathcal{P}_B$ (respectant les pointeurs) qui préserve la longueur des parties et telle que si $s \leq t$ alors $f(s) \leq f(t)$.
- Un *a-morphisme* (ou morphisme d'arènes) de $A$ dans $B$ est une fonction $f$ injective de $M_A$ dans $M_B$ telle que :
  - $\lambda_B \circ f = \lambda_A$
  - $\vdash_B f(m)$ *ssi* $\vdash_A m$
  - $f(m) \vdash_B f(n)$ *ssi* $m \vdash_A n$

  et telle que la fonction induite sur les ensembles de parties soit un $p$-morphisme (il suffit de vérifier que l'image d'une partie est bien une partie).

Un $x$-isomorphisme entre $A$ et $B$ est un $x$-morphisme de $A$ dans $B$ possédant un $x$-morphisme inverse (ce qui entraîne que la fonction sous-jacente est une bijection). On note $A \simeq_x B$ s'il existe un $x$-isomorphisme entre $A$ et $B$.

On dit qu'un connecteur est $x$-associatif, $x$-commutatif, ... si l'isomorphisme sous-jacent est un $x$-isomorphisme.

3.12. Proposition (Propriétés structurelles)
- $\odot$ *et* & *sont* $a$*-commutatifs et* $a$*-associatifs :*

$$A \odot B \simeq_a B \odot A$$
$$A \odot (B \odot C) \simeq_a (A \odot B) \odot C$$
$$A \,\&\, B \simeq_a B \,\&\, A$$
$$A \,\&\, (B \,\&\, C) \simeq_a (A \,\&\, B) \,\&\, C$$

- $\top$ *est* $a$*-neutre pour* $\odot$ *et* & *:*

$$A \odot \top \simeq_a A$$
$$A \,\&\, \top \simeq_a A$$

Démonstration :
- Les $a$-associativités et $a$-commutativités sont immédiates.
- Les $a$-neutralités de $\top$ viennent directement du fait que, pour tout ensemble $A$, $A + \emptyset \simeq A$. □

## 3.2 Stratégies

3.13. Définition (Stratégie)
Soit $A$ un jeu, une *stratégie* $\sigma$ sur le jeu $A$ est un ensemble non vide de parties de $A$ terminant par des coups $J$ et clos par $J$*-préfixe*, tel que :
- $\sigma$ est *déterministe* : si $sm^J \in \sigma$ et $sn^J \in \sigma$ alors $m = n$.
- $\sigma$ est *innocente* : si $smn^J \in \sigma$, $t \in \sigma$, $tm \in \mathcal{P}_A$ et $\ulcorner sm \urcorner = \ulcorner tm \urcorner$ alors $tmn \in \sigma$.

3.14. Lemme (Image d'une stratégie)
*Soient* $\sigma$ *une stratégie sur le jeu* $A$ *et* $f$ *un* $p$-morphisme *de* $A$ *dans* $B$*,* $f(\sigma)$ *est une stratégie sur* $B$*.*

Démonstration : Si $s \leq^J t$ alors $f(s) \leq^J f(t)$ donc $f(\sigma)$ est close par $J$-préfixe. La fonction $f$ est injective donc $f(\sigma)$ est déterministe. Si $f(smn^J) \in f(\sigma)$, $f(t) \in f(\sigma)$, $f(tm) \in \mathcal{P}_B$ et $\ulcorner f(sm) \urcorner = \ulcorner f(tm) \urcorner$ alors $\ulcorner sm \urcorner = \ulcorner tm \urcorner$ et $tmn \in \mathcal{P}_A$ par innocence de $\sigma$ donc $f(tmn) \in \mathcal{P}_B$ et $f(\sigma)$ est innocente. □

3.15. Lemme ($a$-stratégie)
*Soit* $f$ *un* $a$-morphisme *de* $A$ *dans* $B$*,* $\sigma_f = \{s \in \mathcal{P}^J_{A \multimap B} \mid \forall t \leq^J s, t \restriction_A \in \mathcal{P}_A \wedge t \restriction_B = f(t \restriction_A)\}$ *est une stratégie sur* $A \multimap B$*.*

Démonstration : L'ensemble $\sigma_f$ est clos par $J$-préfixe par définition. Si $sm^J, sn^J \in \sigma_f$, supposons que $m \in A$, on a $s \restriction_B = sm \restriction_B = f(sm \restriction_A)$ donc $|s \restriction_B| = |s \restriction_A| + 1$ d'où $n \in A$ (sinon $sn \restriction_B = f(sn \restriction_A)$ entraînerait $s \restriction_B n = f(s \restriction_A)$ ce qui contredirait la propriété sur la longueur) et $f(sn \upharpoonright_A) = sn \upharpoonright_B = s \upharpoonright_B = sm \upharpoonright_B = f(sm \upharpoonright_A)$ donc $sn \upharpoonright_A = sm \upharpoonright_A$ d'où $m = n$ et $\sigma_f$ est déterministe. On montre également que $\sigma_f$ est innocente (en fait sans mémoire) : si $smn^J \in \sigma_f$, $t \in \sigma_f$, $tm \in \mathcal{P}_A$ et si $n \in A$ alors $m = f(n) \in B$ puisque $smn \in \sigma_f$ et $f(tmn \upharpoonright_A) = f(t \upharpoonright_A) f(n) = t \upharpoonright_B f(n) = t \upharpoonright_B m = tmn \upharpoonright_B$ d'où $tmn \in \sigma_f$. □

3.16. Définition (Identité)
Soit $A$ un jeu, la stratégie *identité* $id_A$ sur $A \multimap A$ est la stratégie obtenue par le lemme 3.15 pour le $a$-isomorphisme identité entre $A$ et $A$.

3.17. Définition (Suite d'interaction)
L'ensemble des *suites d'interaction* $\mathrm{int}(A, B, C)$ est l'ensemble des suites $s$ sur $A + B + C$ ayant des pointeurs pour chaque coup excepté ceux initiaux dans $C$ telles que $s \upharpoonright_{A \Rightarrow B} \in \mathcal{P}_{A \multimap B}$ et $s \upharpoonright_{B \Rightarrow C} \in \mathcal{P}_{B \multimap C}$. La suite $s \upharpoonright_{A \multimap C}$ est obtenue en remplaçant le pointeur de chaque coup $a$ dans $A$ pointant sur un coup $b$ dans $B$ (ce qui implique que $a$ est initial dans $A$ et $b$ est initial dans $B$ par définition de $A \multimap B$) par le coup $c$ vers lequel pointe ce coup $b$ de $B$ (qui est nécessairement un coup initial dans $C$), voir [McC96] pour plus de détails.

3.18. Définition (Composition)
Soient $\sigma : A \multimap B$ et $\tau : B \multimap C$ deux stratégies, la *composition* $\sigma; \tau$ de $\sigma$ et $\tau$ est la stratégie sur $A \multimap C$ définie par :

$$\sigma; \tau = \{ s \upharpoonright_{A \multimap C} \in \mathcal{P}_{A \multimap C} \mid s \in \mathrm{int}(A, B, C) \wedge s \upharpoonright_{A \multimap B} \in \sigma \wedge s \upharpoonright_{B \multimap C} \in \tau \}$$

3.19. Proposition (Catégorie monoïdale de jeux HO)
*La catégorie dont les objets sont les jeux HO et les morphismes de $A$ dans $B$ sont les stratégies innocentes sur $A \multimap B$ est une* catégorie monoïdale symétrique fermée *(SMCC) avec produits.*

## 3.3 Exponentielles HO

3.20. Définition (Jeux exponentiels)
Soit $A$ un jeu, le jeu $\sharp A$ a pour arène celle de $A$ et $\mathcal{P}_{\sharp A} = \{ s \in L_A \mid \forall m \text{ initial}, s \upharpoonright_m \in \mathcal{P}_A \}$.

3.21. Remarque : La propriété (i) de la définition 3.8 entraîne que toute partie de $A$ est une partie de $\sharp A$.

3.22. Proposition (Propriétés structurelles)
- $\sharp(A \,\&\, B) \simeq_a \sharp A \odot \sharp B$.
- $\sharp\top = \top$.

Démonstration :
- Les arènes de $\sharp(A \,\&\, B)$ et $\sharp A \odot \sharp B$ sont les mêmes, l'identité est donc une fonction entre les coups de ces deux jeux qui vérifie les conditions requises pour être un $a$-isomorphisme, il reste toutefois à vérifier que la fonction induite sur les ensembles de parties est un $p$-isomorphisme (ce qui est faux pour les arènes $A \,\&\, B$ et $A \odot B$). Soit $s$ une partie du jeu $\sharp(A \,\&\, B)$, soit $m$ un coup initial de $s \upharpoonright_{\sharp A}$ $(= s \upharpoonright_A)$, on a $s \upharpoonright_A \upharpoonright_m = s \upharpoonright_m$ or $m$ est un coup initial de $s$ donc $s \upharpoonright_m \in \mathcal{P}_{A \& B}$ mais $m \in M_A$ donc c'est une partie de $A$. De même, si $n$ est un coup initial de $s \upharpoonright_B$, $s \upharpoonright_B \upharpoonright_n$ est une partie de $B$ donc $s$ est une partie de $\sharp A \odot \sharp B$. Réciproquement, si $s$ est une partie de $\sharp A \odot \sharp B$, et si $m$ est un coup initial de $s$, supposons que $m \in M_A$, $s \upharpoonright_m \upharpoonright_A = s \upharpoonright_A \upharpoonright_m$ est une partie de $A$ (et de même si $m \in M_B$).

– L'arène de $\sharp\top$ est vide, or il n'y a qu'un tel jeu donc $\sharp\top = \top$. $\square$

3.23. Définition (Stratégies exponentielles)
Soit $A$ un jeu *bien ouvert*, on définit les stratégies suivantes :

(déréliction) $d_A = id_A : \sharp A \multimap A$ bien définie d'après la remarque 3.21.

(contraction) Si $s$ est une partie de $\sharp A_0 \multimap \sharp A_1 \odot \sharp A_2$ (où les indices ne servent qu'à distinguer les occurrences) et si $I$ est l'ensemble de ses coups initiaux dans $A_1$ et $J$ ceux dans $A_2$, on note $s_1 = s \upharpoonright_I$ et $s_2 = s \upharpoonright_J$. On peut alors définir la stratégie $c_A = \{s \in \mathcal{P}^J_{\sharp A_0 \multimap \sharp A_1 \odot \sharp A_2} \mid \forall t \leq^J s, t_1 \upharpoonright_{\sharp A_0} = t_1 \upharpoonright_{\sharp A_1} \wedge t_2 \upharpoonright_{\sharp A_0} = t_2 \upharpoonright_{\sharp A_2}\} : \sharp A_0 \multimap \sharp A_1 \odot \sharp A_2$.

(affaiblissement) $w_A : \sharp A \multimap \top$ est la stratégie $\{\varepsilon\}$.

(promotion) Soit $\sigma : \sharp A \multimap B$ une stratégie, la stratégie $\sigma^\dagger : \sharp A \multimap \sharp B$ est définie par $\sigma^\dagger = \{s \in L_{\sharp A \multimap \sharp B} \mid \forall m \text{ initial}, s \upharpoonright_m \in \sigma\}$.

3.24. Définition (Composition (bis))
Soit $\sigma : \sharp A \multimap B$ et $\tau : \sharp B \multimap C$ deux stratégies, on peut les composer pour obtenir une stratégie dans $\sharp A \multimap C$ par $\sigma^\dagger ; \tau$.

3.25. Proposition (Catégorie cartésienne de jeux HO)
*La catégorie dont les objets sont les jeux HO bien ouverts et les morphismes de $A$ dans $B$ sont les stratégies innocentes sur $\sharp A \multimap B$ est une* catégorie cartésienne fermée *(CCC).*

3.26. Lemme (!-lemme)
*Soit $B$ un jeu bien ouvert et soit $\sigma : \sharp A \multimap \sharp B$ une stratégie, on a $\sigma = (\sigma; d_B)^\dagger$.*

3.27. Remarque : Ce lemme est un résultat de réversibilité du connecteur $\sharp$ dans la sémantique.

## 3.4 Jeux AJM

3.28. Définition (Arène AJM)
Une *arène AJM $A$* est un couple $(M_A, \lambda_A)$ où :
- $M_A$ est un ensemble appelé ensemble de *coups*,
- $\lambda_A$ est une fonction de $M_A$ dans $\{O, J\}$.

3.29. Définition (Jeu AJM)
Un *jeu AJM $A$* est un quadruplet $(M_A, \lambda_A, \mathcal{P}_A, \equiv_A)$ où $(M_A, \lambda_A)$ est une arène AJM et $\mathcal{P}_A$, appelé ensemble des *parties* de $A$, est un ensemble de suites alternées de coups de $A$ commençant par un coup $O$, enfin $\equiv_A$ est une relation d'équivalence entre parties de même longueur telle que :
- $s \equiv_A t \wedge s' \leq s \wedge t' \leq t \wedge |s'| = |t'| \Rightarrow s' \equiv_A t'$
- $s \equiv_A t \wedge sm \in \mathcal{P}_A \Rightarrow \exists n (tn \in \mathcal{P}_A \wedge sm \equiv_A tn)$

3.30. Définition (Constructions linéaires)

($\top$) On note $\top$ l'unique jeu dont l'ensemble de coups est vide : $(\emptyset, \emptyset, \{\varepsilon\}, \{\varepsilon \equiv_\top \varepsilon\})$.

($\odot$) Si $A$ et $B$ sont deux jeux, le jeu $A \odot B$ est construit comme dans le cas HO avec comme relation d'équivalence : $s \equiv_{A \odot B} t$ si $s \upharpoonright_A \equiv_A t \upharpoonright_A$, $s \upharpoonright_B \equiv_B t \upharpoonright_B$ et l'*entrelacement* des parties est le même (*i.e.* le $i^{\text{e}}$ coup de $s$ est dans $A$ *ssi* le $i^{\text{e}}$ coup de $t$ est dans $A$).

($\multimap$) Idem.

(&) Si $A$ et $B$ sont deux jeux, le jeu $A \& B$ est construit comme dans le cas HO avec comme relation d'équivalence : $s \equiv_{A \& B} t$ si $s \equiv_A t$ avec $s$ et $t$ dans $\mathcal{P}_A$ ou si $s \equiv_B t$ avec $s$ et $t$ dans $\mathcal{P}_B$.

3.31. Remarque : La notion d'entrelacement de parties étant suffisamment intuitive, on ne rentrera pas dans une définition technique très précise. Une partie $u$ est un entrelacement de deux parties $s$ et $t$, si tout coup de $u$ est soit un coup de $s$ soit un coup de $t$, si réciproquement tout coup de $s$ ou de $t$ est dans $u$ et si l'ordre des coups de $s$ et de $t$ est respecté dans $u$ ($s$ et $t$ sont donc des sous-suites de $u$). Si $u$ (resp. $u'$) est un entrelacement de $s$ et $t$ (resp. $s'$ et $t'$), on dit que $u$ et $u'$ ont même entrelacement si : le $i^{\mathrm{e}}$ coup de $u$ est un coup de $s$ *ssi* le $i^{\mathrm{e}}$ coup de $u'$ est un coup de $s'$ (ce qui entraîne que le $i^{\mathrm{e}}$ coup de $u$ est un coup de $t$ *ssi* le $i^{\mathrm{e}}$ coup de $u'$ est un coup de $t'$).

La notion de stratégie est la même que pour les jeux HO, excepté que la condition d'innocence est remplacée par le fait que les stratégies doivent être *sans mémoire* :
– $smn^J \in \sigma \wedge t \in \sigma \wedge tm \in \mathcal{P}_A \Rightarrow tmn^J \in \sigma$

Les constructions sur les stratégies sont les mêmes que pour les jeux HO en oubliant les informations de pointeurs.

3.32. Proposition (Catégorie monoïdale de jeux AJM)
*La catégorie dont les objets sont les jeux AJM et les morphismes de $A$ dans $B$ sont les stratégies sans mémoire sur $A \multimap B$ est une* catégorie monoïdale symétrique fermée*.*

3.33. Définition (Équivalence partielle de stratégies)
Deux stratégies $\sigma$ et $\tau$ sur un jeu $A$ sont *équivalentes*, noté $\sigma \approx \tau$, si la condition suivante est vérifiée :

$$smn^J \in \sigma \wedge t \in \tau \wedge sm \equiv_A tm' \Rightarrow \exists n'(tm'n' \in \tau \wedge smn \equiv_A tm'n')$$

ainsi que la condition symétrique.

Cette relation entre stratégies est une relation d'équivalence partielle et on ne s'intéressera désormais qu'aux stratégies auto-équivalentes.

3.34. Définition (Jeux exponentiels AJM)
Soit $A$ un jeu, le jeu AJM $\sharp A$ est le quadruplet $(\mathbb{N} \times M_A, (i,a) \mapsto \lambda_A(a), \mathcal{P}_{\sharp A}, \equiv_{\sharp A})$ où $\mathcal{P}_{\sharp A} = \{s \mid \forall i \in \mathbb{N}, s \upharpoonright_i \in \mathcal{P}_A\}$ (où $s \upharpoonright_i$ est la sous-suite de $s$ contenant les coups d'indice $i$) et $s \equiv_{\sharp A} t$ s'il existe une permutation des entiers $\theta$ telle que pour tout $i \in \mathbb{N}$, $s \upharpoonright_i \equiv_A t \upharpoonright_{\theta(i)}$ et $s$ et $\theta(t)$ ont même entrelacement (où $\theta(t)$ est obtenue est remplaçant l'indice $i$ par $\theta(i)$ dans chaque coup de $t$).

3.35. Définition (Stratégies exponentielles AJM)
Soit $A$ un jeu AJM, soit $k$ un entier, soient $l$ et $r$ deux injections de $\mathbb{N}$ dans $\mathbb{N}$ à images disjointes et soit $p$ une injection de $\mathbb{N} \times \mathbb{N}$ dans $\mathbb{N}$, on définit les stratégies suivantes :

(déréliction) $d_A = \{s \in \mathcal{P}^J_{\sharp A \multimap A} \mid \forall t \leq^J s, t \upharpoonright_{\sharp A} = k.t \upharpoonright_A\} : \sharp A \multimap A$, changer d'indice $k$ donne lieu à une stratégie équivalente.

(contraction) $c_A = \{s \in \mathcal{P}^J_{\sharp A_0 \multimap \sharp A_1 \odot \sharp A_2} \mid \forall t \leq^J s, \forall i \in \mathbb{N}, t \upharpoonright_{\sharp A_0} \upharpoonright_{l(i)} = t \upharpoonright_{\sharp A_1} \upharpoonright_i \wedge t \upharpoonright_{\sharp A_0} \upharpoonright_{r(i)} = t \upharpoonright_{\sharp A_2} \upharpoonright_i\} : \sharp A_0 \multimap \sharp A_1 \odot \sharp A_2$, changer de paire d'injections $(l,r)$ donne également lieu à une stratégie équivalente.

(affaiblissement) $w_A : \sharp A \multimap \top$ est la stratégie $\{\varepsilon\}$.

(promotion) Soit $\sigma : \sharp A \multimap B$ une stratégie, la stratégie $\sigma^\dagger : \sharp A \multimap \sharp B$ est obtenue en composant la stratégie $\{s \in \mathcal{P}^J_{\sharp A \multimap \sharp\sharp A} \mid \forall t \leq^J s, t \upharpoonright_{\sharp A} = p(t \upharpoonright_{\sharp\sharp A})\} : \sharp A \multimap \sharp\sharp A$ (où si $s$ est une suite de coups de $\sharp\sharp A$, $p(s)$ est obtenue en remplaçant chaque coup $(i,(j,m))$ par $(p(i,j),m))$ et la stratégie $!\sigma = \{s \in \mathcal{P}^J_{\sharp\sharp A \multimap \sharp B} \mid \forall i \in \mathbb{N}, s \upharpoonright_i \in \sigma\} : \sharp\sharp A \multimap \sharp B$. Prendre une autre fonction de codage $p$ donne une stratégie équivalente.

3.36. Proposition (Catégorie cartésienne de jeux AJM)
*La catégorie dont les objets sont les jeux AJM et les morphismes de $A$ dans $B$ sont les classes de stratégies sans mémoire auto-équivalentes sur $\sharp A \multimap B$ est une* catégorie cartésienne fermée.

3.37. Définition (Jeu bien ouvert)
Un jeu AJM est *bien ouvert* si le premier coup d'une partie n'apparaît jamais au cours d'une partie.

3.38. Remarque : Attention, cette notion de bonne ouverture pour les jeux AJM est différente de celle des jeux HO.

3.39. Lemme (!-lemme)
*Soit $A$ un jeu bien ouvert et soit $\sigma : \sharp A \multimap \sharp B$ une stratégie, on a $\sigma \approx (\sigma; d_B)^\dagger$.*

# Deuxième partie

# Logique Linéaire Polarisée

Cette partie développe une théorie de la démonstration polarisée. Après avoir défini le système LLP, nous aborderons, dans le cadre polarisé, toutes les branches syntaxiques et sémantiques de l'étude habituelle de la logique linéaire : calcul des séquents, réseaux de preuve, sémantique cohérente, sémantique catégorique, sémantique des jeux et géométrie de l'interaction. Ceci permettra en particulier d'étendre des notions jusqu'ici limitées à de plus petits systèmes que LLP.

# Chapitre 4

# Le système LLP

L'introduction, dans LL, de la notion de polarités définie dans LC [Gir91a] et issue des travaux de J.-M. Andreoli sur la *focalisation* [And90], va nous permettre de définir le système LLP. Nous étudions ici les principales propriétés logiques de ce système pour la plupart provenant des liens entre négatif et réversibilité et entre positif et focalisation.

## 4.1 Calcul des séquents

Nous allons montrer comment s'effectue la construction du calcul des séquents de LLP à partir de l'introduction des polarités dans LL.

### 4.1.1 Formules polarisées

4.1. Définition (Formule polarisée)
Une *formule polarisée* est une formule de LL engendrée par la grammaire suivante :

$$\begin{array}{lclclclclclclcl} P & ::= & X & \mid & P \otimes P & \mid & P \oplus P & \mid & 1 & \mid & 0 & \mid & \exists X P & \mid & !N \\ N & ::= & X^{\perp} & \mid & N \parr N & \mid & N \,\&\, N & \mid & \perp & \mid & \top & \mid & \forall X N & \mid & ?P \end{array}$$

Les connecteurs $\parr$, $\&$, $\perp$, $\top$ et $\forall$ (resp. $\otimes$, $\oplus$, $1$, $0$ et $\exists$) sont dits *négatifs* (resp. *positifs*).

Notations : $N$ et $M$ désigneront des formules négatives et $P$ et $Q$ des formules positives. $\Gamma$ et $\Delta$ seront des multi-ensembles de formules et $\mathcal{N}$ un multi-ensemble de formules négatives.

4.2. Remarque : Nous ne considérons ici que la quantification du *second ordre*. Les *quantificateurs du premier ordre* se traiteraient de manière extrêmement similaire mais nécessiteraient d'alourdir le langage. Il en sera de même pour le reste de cette thèse : réseaux de preuve, logique classique, traductions, ...

4.3. Lemme (Formule positive)
*Soit $\pi$ une preuve de LL ne contenant que des formules polarisées et n'utilisant pas la règle $\top$, si $\vdash \Gamma$ est la conclusion de $\pi$ alors $\Gamma$ contient au plus une formule positive.*

Démonstration : Par récurrence sur la taille de $\pi$, en considérant chaque cas de dernière règle pour $\pi$ :

($ax$) $\Gamma = P, P^{\perp}$ contient exactement une formule positive.

($cut$) Les deux prémisses contiennent chacune au plus une formule positive par hypothèse de récurrence. Or, parmi les deux formules duales coupées, l'une est nécessairement positive donc la conclusion de la règle ne contient qu'au plus une formule positive.

($\parr$) Cette règle ne modifie pas le nombre de formules positives.

($\otimes$) Pour que la règle puisse s'appliquer chaque prémisse doit contenir une formule positive donc, par hypothèse de récurrence, exactement une. La conclusion contient une unique formule positive qui est le tenseur de celles des prémisses.

($\&$) Si les prémisses contiennent une formule positive, c'est la même et elle est dans le contexte. Par contraction du contexte on obtient alors au plus une formule positive dans la conclusion.

($\oplus_i$) Pour que la règle s'applique et par hypothèse de récurrence, la prémisse contient exactement une formule positive qui donne une formule positive dans la conclusion.

($\bot$) Cette règle ajoute une formule négative sans toucher aux formules positives.

(1) $\Gamma = 1$ contient une formule positive.

($\forall$) Cette règle ne change rien au nombre de formules positives et négatives.

($\exists$) Comme pour la règle $\oplus_i$.

($?d$) Pour que la règle s'applique, la prémisse doit contenir une formule positive $P$ qui disparaît en $?P$ dans la conclusion.

($?c$) Comme pour la règle $\parr$.

($?w$) Comme pour la règle $\bot$.

($!$) Les formules du contexte sont nécessairement négatives et on obtient une formule $!N$ positive. $\square$

4.4. Définition ($\mathsf{LL}_{\text{pol}}$)
Le système $\mathsf{LL}_{\text{pol}}$ est le fragment de $\mathsf{LL}$ constitué uniquement des formules polarisées et dont la règle $\top$ n'introduit qu'au plus une formule positive.

4.5. Remarque : Par *fragment* d'un système logique, on entendra sous-système défini par restriction de l'espace des formules en ne modifiant pas les règles. La contrainte sur les formules doit bien entendu être stable par restriction à une sous-formule. Cependant le cas de la règle $\top$ est particulier, en effet son rôle est que $\top$ soit l'élément neutre de $\&$. Conserver la règle de $\mathsf{LL}$ autorisant un contexte quelconque n'est pas forcément nécessaire pour cela et peut faire perdre des propriétés du système. Ainsi dans le cas de $\mathsf{LL}_{\text{pol}}$, la propriété donnée par le lemme 4.3 est cruciale comme nous le verrons à de nombreux endroits. Il est naturel de contraindre la règle $\top$ à vérifier cette propriété, dans la mesure où l'on peut toujours prouver $(\top \,\&\, N) \multimapboth N$.

4.6. Définition ($\mathsf{LL}^+_{\text{pol}}$)
$\mathsf{LL}^+_{\text{pol}}$ est le fragment de $\mathsf{LL}$ constitué des formules *faiblement polarisées* (formules faiblement positives et formules faiblement négatives) :

$$\begin{array}{lcccccccccccccccccc} P & ::= & X & \mid & P \otimes P & \mid & P \oplus P & \mid & 1 & \mid & 0 & \mid & \exists X P & \mid & !N & \mid & !P \\ N & ::= & X^\bot & \mid & N \parr N & \mid & N \,\&\, N & \mid & \bot & \mid & \top & \mid & \forall X N & \mid & ?P & \mid & ?N \end{array}$$

et dont la règle $\top$ n'introduit qu'au plus une formule positive.

Une formule polarisée dont les atomes positifs et négatifs sont sous la portée d'un connecteur exponentiel est dite *à polarité explicite*.

4.7. Lemme (Préservation du type)
*Soit $N$ une formule négative à polarité explicite, $\vdash !N^\perp, N$ est prouvable dans $\mathsf{LL}^+_{\mathrm{pol}}$.*

Démonstration : Par récurrence sur la taille de $N$ en considérant chaque cas de connecteur principal (le cas de l'atome étant exclu par hypothèse) :

($\bot$) Prouvons $\vdash !1, \bot$ :

$$\dfrac{\dfrac{\dfrac{}{\vdash 1}\,1}{\vdash !1}\,!}{\vdash !1, \bot}\,\bot$$

($\top$) $\vdash !0, \top$ se démontre par une simple règle $\top$.

($\parr$) Si $N = N_1 \parr N_2$, par hypothèse de récurrence, $\vdash !N_1^\perp, N_1$ et $\vdash !N_2^\perp, N_2$ sont prouvables.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash N_1^\perp, N_1}\,ax \qquad \dfrac{}{\vdash N_2^\perp, N_2}\,ax}{\vdash N_1^\perp \otimes N_2^\perp, N_1, N_2}\,\otimes}{\vdash N_1^\perp \otimes N_2^\perp, ?N_1, N_2}\,?d}{\vdash N_1^\perp \otimes N_2^\perp, ?N_1, ?N_2}\,?d}{\vdash !(N_1^\perp \otimes N_2^\perp), ?N_1, ?N_2}\,! \qquad \vdash !N_1^\perp, N_1}{\vdash !(N_1^\perp \otimes N_2^\perp), N_1, ?N_2}\,cut \qquad \vdash !N_2^\perp, N_2}{\vdash !(N_1^\perp \otimes N_2^\perp), N_1, N_2}\,cut}{\vdash !(N_1^\perp \otimes N_2^\perp), N_1 \parr N_2}\,\parr$$

($\&$) Si $N = N_1 \& N_2$, par hypothèse de récurrence, $\vdash !N_1^\perp, N_1$ et $\vdash !N_2^\perp, N_2$ sont prouvables.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash N_1^\perp, N_1}\,ax}{\vdash N_1^\perp \oplus N_2^\perp, N_1}\,\oplus_1}{\vdash N_1^\perp \oplus N_2^\perp, ?N_1}\,?d}{\vdash !(N_1^\perp \oplus N_2^\perp), ?N_1}\,! \qquad \vdash !N_1^\perp, N_1}{\vdash !(N_1^\perp \oplus N_2^\perp), N_1}\,cut \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash N_2^\perp, N_2}\,ax}{\vdash N_1^\perp \oplus N_2^\perp, N_2}\,\oplus_2}{\vdash N_1^\perp \oplus N_2^\perp, ?N_2}\,?d}{\vdash !(N_1^\perp \oplus N_2^\perp), ?N_2}\,! \qquad \vdash !N_2^\perp, N_2}{\vdash !(N_1^\perp \oplus N_2^\perp), N_2}\,cut}{\vdash !(N_1^\perp \oplus N_2^\perp), N_1 \& N_2}\,\&$$

($\forall$) Si $N = \forall X N_1$, par hypothèse de récurrence, $\vdash !N_1^\perp, N_1$ est prouvable.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash N_1^\perp, N_1}\,ax}{\vdash \exists X N_1^\perp, N_1}\,\exists}{\vdash \exists X N_1^\perp, ?N_1}\,?d}{\vdash !\exists X N_1^\perp, ?N_1}\,! \qquad \vdash !N_1^\perp, N_1}{\vdash !\exists X N_1^\perp, N_1}\,cut}{\vdash !\exists X N_1^\perp, \forall X N_1}\,\forall$$

($?$) Prouvons $\vdash !!P^\perp, ?P$ :

$$\dfrac{\dfrac{}{\vdash !P^\perp, ?P}\,ax}{\vdash !!P^\perp, ?P}\,!$$

$\square$

4.8. Lemme (Règles structurelles polarisées)
*Soient $N$ une formule négative et $\mathcal{N}$ un multi-ensemble de formules négatives tous les deux à polarités explicites. Les règles suivantes sont prouvables dans $\mathsf{LL}^{+}_{\mathrm{pol}}$ :*

$$\frac{\vdash \Gamma}{\vdash \Gamma, N} \qquad \frac{\vdash \Gamma, N, N}{\vdash \Gamma, N} \qquad \frac{\vdash \mathcal{N}, N}{\vdash \mathcal{N}, !N}$$

Démonstration : On utilise essentiellement le lemme 4.7 en introduisant des coupures :

$$\dfrac{\dfrac{\vdash \Gamma}{\vdash \Gamma, ?N}\,?w \qquad \vdash !N^{\perp}, N}{\vdash \Gamma, N}\,cut \qquad \dfrac{\dfrac{\dfrac{\dfrac{\vdash \Gamma, N, N}{\vdash \Gamma, ?N, N}\,?d}{\vdash \Gamma, ?N, ?N}\,?d}{\vdash \Gamma, ?N}\,?c \qquad \vdash !N^{\perp}, N}{\vdash \Gamma, N}\,cut$$

$$\dfrac{\dfrac{\vdash N, ?N_1, ..., ?N_k}{\vdash !N, ?N_1, ..., ?N_k}\,! \qquad \vdash !N_1^{\perp}, N_1}{\vdash !N, N_1, ?N_2, ..., ?N_k}\,cut \quad \text{(où } \vdash N, ?N_1, ..., ?N_k \text{ est obtenu de } \vdash N, N_1, ..., N_k \text{ par } ?d\text{)}$$

$$\vdots$$

$$\dfrac{\vdash !N, N_1, ..., N_{k-1}, ?N_k \qquad \vdash !N_k^{\perp}, N_k}{\vdash !N, N_1, ..., N_k}\,cut$$

□

Le système $\mathsf{LLP}$ est obtenu en ajoutant ces règles à $\mathsf{LL}_{\mathrm{pol}}$ pour toute formule négative. Sur le plan de la prouvabilité, cet enrichissement ne fait qu'ajouter les règles structurelles sur les atomes négatifs ce qui est parfaitement dans l'idée qu'un atome négatif représente une formule négative générique. Puisque celles-ci vérifient les règles structurelles il est normal que ce soit le cas des atomes. Cette extension pouvant ainsi sembler anodine est en fait essentielle du point de vue des preuves (et pas simplement de la prouvabilité) comme cette thèse tendra à le prouver ! Contrairement à $\mathsf{LL}_{\mathrm{pol}}$, $\mathsf{LLP}$ n'est pas un sous-système de $\mathsf{LL}$ mais on a $\mathsf{LL}_{\mathrm{pol}} = \mathsf{LL} \cap \mathsf{LLP}$.

### 4.1.2 Règles

On aboutit au système $\mathsf{LLP}$ dont les règles s'écrivent :

$$\frac{}{\vdash N, N^{\perp}}\,ax \qquad \frac{\vdash \Gamma, N \qquad \vdash N^{\perp}, \Delta}{\vdash \Gamma, \Delta}\,cut$$

$$\frac{\vdash \Gamma, N, M}{\vdash \Gamma, N \mathbin{⅋} M}\,⅋ \qquad \frac{\vdash \Gamma, P \qquad \vdash \Delta, Q}{\vdash \Gamma, \Delta, P \otimes Q}\,\otimes$$

$$\frac{\vdash \Gamma, N \qquad \vdash \Gamma, M}{\vdash \Gamma, N \,\&\, M}\,\& \qquad \frac{\vdash \Gamma, P}{\vdash \Gamma, P \oplus Q}\,\oplus_1 \qquad \frac{\vdash \Gamma, Q}{\vdash \Gamma, P \oplus Q}\,\oplus_2$$

$$\frac{\vdash \mathcal{N}, N}{\vdash \mathcal{N}, !N}\,! \qquad \frac{\vdash \Gamma, P}{\vdash \Gamma, ?P}\,?d \qquad \frac{\vdash \Gamma}{\vdash \Gamma, N}\,?w \qquad \frac{\vdash \Gamma, N, N}{\vdash \Gamma, N}\,?c$$

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

$$\frac{\vdash \Gamma, N}{\vdash \Gamma, \forall X N}\ \forall \qquad\qquad \frac{\vdash \Gamma, P[^{Q}/_{X}]}{\vdash \Gamma, \exists X P}\ \exists$$

Avec les contraintes suivantes :
- le contexte $\Gamma$ de la règle $\top$ contient au plus une formule positive ;
- la variable $X$ n'est pas libre dans le contexte de la règle $\forall$.

4.9. Lemme (Formule positive)
*Soit $\pi$ une preuve de* LLP*, si $\vdash \Gamma$ est la conclusion de $\pi$ alors $\Gamma$ contient au plus une formule positive.*

Démonstration : Comme pour le lemme 4.3 puisque les modifications des règles structurelles ne changent pas la polarité des formules. □

4.10. Remarque : Il est également possible d'expliciter le lemme 4.9 dans les règles comme cela est fait à la section 10.1.3, en indiquant explicitement les formules positives qui peuvent apparaître dans les contextes.

## 4.2 Propriétés

### 4.2.1 Règles réversibles

4.11. Définition (Connecteur réversible)
Un connecteur $\circledast$ est *réversible* si pour toute preuve $\pi$ du séquent $\vdash \Gamma, A$ où $\circledast$ est le connecteur principal de $A$, il existe une preuve sans coupure $\pi'$ de $\vdash \Gamma, A$ dont la dernière règle est une règle d'introduction de $\circledast$.

4.12. Lemme (Connecteurs réversibles de LL)
*Les connecteurs réversibles de* LL *sont $\parr$, $\&$, $\bot$, $\top$ et $\forall$.*

Démonstration : Pour chacun des connecteurs, on définit $\pi_0$ de la manière suivante :

($\parr$)

$$\dfrac{\dfrac{\dfrac{\overset{\pi}{\vdash \Gamma, A \parr B} \qquad \dfrac{\dfrac{}{\vdash A^{\perp}, A}\,ax \qquad \dfrac{}{\vdash B^{\perp}, B}\,ax}{\vdash A^{\perp} \otimes B^{\perp}, A, B}\,\otimes}{\vdash \Gamma, A, B}\,cut}{\vdash \Gamma, A \parr B}\,\parr}{}$$

(&)

$$\dfrac{\dfrac{\overset{\pi}{\vdash \Gamma, A \& B} \qquad \dfrac{\dfrac{}{\vdash A^{\perp}, A}\,ax}{\vdash A^{\perp} \oplus B^{\perp}, A}\,\oplus_1}{\vdash \Gamma, A}\,cut \qquad \dfrac{\overset{\pi}{\vdash \Gamma, A \& B} \qquad \dfrac{\dfrac{}{\vdash B^{\perp}, B}\,ax}{\vdash A^{\perp} \oplus B^{\perp}, B}\,\oplus_2}{\vdash \Gamma, B}\,cut}{\vdash \Gamma, A \& B}\,\&$$

($\bot$)

$$\dfrac{\dfrac{\overset{\pi}{\vdash \Gamma, \bot} \qquad \dfrac{}{\vdash 1}\,1}{\vdash \Gamma}\,cut}{\vdash \Gamma, \bot}\,\bot$$

(⊤)

$$\frac{}{\vdash \Gamma, \top}\top$$

(∀) Avec un éventuel renommage de la variable liée :

$$\dfrac{\dfrac{\pi}{\vdash \Gamma, \forall XA} \qquad \dfrac{\dfrac{}{\vdash A^\perp, A}ax}{\vdash \exists XA^\perp, A}\exists}{\dfrac{\vdash \Gamma, A}{\vdash \Gamma, \forall XA}\forall}cut$$

La preuve $\pi'$ s'obtient à partir de $\pi_0$ en éliminant les coupures ce qui ne modifie pas la dernière règle.

Il reste à montrer que les autres connecteurs ne sont pas réversibles :

(⊗) Ni $\vdash X, X^\perp, Y^\perp$, ni $\vdash Y, X^\perp, Y^\perp$ ne sont prouvables, donc une preuve de $\vdash X \otimes Y, X^\perp \parr Y^\perp$ ne peut pas terminer par une règle ⊗.

(⊕) $\vdash X, Y^\perp$ et $\vdash Y, X^\perp$ ne sont pas prouvables donc une preuve de $\vdash X \oplus Y, X^\perp \& Y^\perp$ ne peut pas terminer par une règle ⊕.

(1) $\vdash 1, \perp$ n'est pas prouvable par la règle 1.

(0) Pas de règle d'introduction pour le connecteur 0.

(∃) Toute preuve sans coupure de $\vdash \exists XX, \forall XX^\perp$ termine par une règle *ax* ou une règle ∀.

(?) $\vdash X, !X^\perp$ n'est pas prouvable donc une preuve de $\vdash ?X, !X^\perp$ ne peut pas se terminer par une règle $?d$.

(!) Une preuve de $\vdash \perp, !\top$ ne peut pas se terminer par une règle ! puisque le contexte devrait être de la forme $?\Gamma$. □

4.13. Remarques :
- La propriété utilisée pour montrer la réversibilité d'un connecteur ⊛ est l'existence d'une *η-expansion* de l'identité pour les formules de connecteur principal ⊛ qui termine par une règle d'introduction de ⊛.
- La non réversibilité du connecteur ! est un peu particulière, en effet elle n'est vraie que parce la règle ! nécessite une contrainte forte sur le contexte. On a le résultat plus faible suivant : si $\vdash ?\Gamma, !A$ est prouvable dans LL, il existe une preuve sans coupure de $\vdash ?\Gamma, !A$ dont la dernière règle est une règle !. En particulier le lemme suivant montre que ! devient réversible dans LLP.
- Ainsi les connecteurs réversibles sont les connecteurs négatifs et (selon le cas) le connecteur ! ce qui souligne bien que la polarité de ce connecteur n'est pas évidente.

4.14. Lemme (Connecteurs réversibles de LLP)
*Les connecteurs réversibles de* LLP *sont* $\parr$, &, ⊥, ⊤, ∀ *et* !.

Démonstration : La réversibilité des connecteurs $\parr$, &, ⊥, ⊤ et ∀ se prouve exactement de la même manière que dans le lemme précédent ainsi que la non réversibilité des connecteurs ⊗, ⊕, 1, 0, ∃ et ?.

Il reste à montrer la réversibilité du connecteur !. Soit $\pi$ une preuve de $\vdash \Gamma, !N$ dans LLP. Par le lemme 4.9, $\Gamma$ ne contient que des formules négatives, on peut donc construire la preuve suivante :

$$\dfrac{\dfrac{\pi}{\vdash \Gamma, !N} \quad \dfrac{\dfrac{}{\vdash N^{\perp}, N}\, ax}{\vdash ?N^{\perp}, N}\, ?d}{\dfrac{\vdash \Gamma, N}{\vdash \Gamma, !N}\, !}\, cut$$

On obtient le résultat cherché en éliminant les coupures. □

4.15. Remarque : On peut également définir une notion plus forte de réversibilité relativement à une sémantique donnée $[\![.]\!]$ en demandant dans la définition 4.11 que $[\![\pi]\!] = [\![\pi']\!]$. Les deux lemmes ci-dessus restent vrais si on considère par exemple la réversibilité par rapport à la sémantique cohérente.

Ceci montre que la transformation de $\pi$ en $\pi'$, qui correspond à certaines permutations de règles, est sans conséquence. En particulier certaines de ces permutations seront réalisées par les réseaux de preuve.

### 4.2.2 Renversement

Comme on l'a déjà remarqué, LLP n'est pas un sous-système de LL. Pourtant le lemme 4.8 montre que les règles ajoutés dans LLP sont (presque toujours) démontrables dans LL. Ceci suggère une traduction simple de LLP dans LL qui consiste à remplacer chaque règle structurelle généralisée par sa preuve dans LL. Cette traduction $(.)^{\rho}$ est appelée *renversement*. Elle a initialement été étudiée pour la logique classique [DJS97, QTdF96].

4.16. Définition (Traduction par renversement)
Les formules polarisées sont traduites par $(.)^{\rho}$ en ajoutant les connecteurs exponentiels appropriés sur les atomes. Ce qui donne pour les formules positives :

$$\begin{array}{ccc}
X & \rightsquigarrow & !X \\
P_1 \otimes P_2 & \rightsquigarrow & P_1^{\rho} \otimes P_2^{\rho} \\
P_1 \oplus P_2 & \rightsquigarrow & P_1^{\rho} \oplus P_2^{\rho} \\
1 & \rightsquigarrow & 1 \\
0 & \rightsquigarrow & 0 \\
\exists X P & \rightsquigarrow & \exists X P^{\rho} \\
!N & \rightsquigarrow & !N^{\rho}
\end{array}$$

et pour les formules négatives, $N^{\rho} = ((N^{\perp})^{\rho})^{\perp}$.

Soit $\pi$ une preuve de $\vdash \Gamma$ dans LLP, la preuve $\pi^{\rho}$ de $\vdash \Gamma^{\rho}$ dans $\mathsf{LL}_{\mathrm{pol}}^{+}$ est obtenue en remplaçant chaque règle structurelle généralisée de $\pi$ par la démonstration correspondante donnée au lemme 4.8, puisque les formules obtenues par la traduction $(.)^{\rho}$ sont à polarité explicite.

L'effet géométrique de cette traduction, lorsque l'on élimine les coupures introduites, est de faire remonter les règles structurelles à travers les connecteurs logiques négatifs (voir [DJS97]). Bien que cette opération puisse paraître anodine, elle cache en fait une réelle transformation des preuves qui correspond à une sorte de CPS-traduction comme nous le montrerons à la section 13.3.

### 4.2.3 Conservation

4.17. Lemme (Prouvabilité dans $\mathsf{LL}_{\mathrm{pol}}^{+}$ et LLP)
*Le séquent* $\vdash \Gamma^{\rho}$ *est prouvable dans* $\mathsf{LL}_{\mathrm{pol}}^{+}$ ssi *le séquent* $\vdash \Gamma$ *est prouvable dans* LLP.

Démonstration : Soit $\pi$ une preuve de $\vdash \Gamma^\rho$ dans $\mathsf{LL}^+_{\mathrm{pol}}$, en remplaçant les $!X$ par $X$ et les $?X^\perp$ par $X^\perp$, on obtient une preuve de $\vdash \Gamma$ dans $\mathsf{LLP}$. Soit $\pi$ une preuve de $\vdash \Gamma$ dans $\mathsf{LLP}$, $\pi^\rho$ est une preuve de $\vdash \Gamma^\rho$ dans $\mathsf{LL}^+_{\mathrm{pol}}$. □

4.18. Lemme (Prouvabilité dans $\mathsf{LL}_{\mathrm{pol}}$ et $\mathsf{LLP}$)
*Le séquent $\vdash \Gamma$ sans atome est prouvable dans $\mathsf{LL}_{\mathrm{pol}}$* ssi *il est prouvable dans $\mathsf{LLP}$.*

Démonstration : Puisque $\mathsf{LL}_{\mathrm{pol}} \subset \mathsf{LLP}$, la première implication est immédiate. Pour la deuxième, par le lemme 4.17, $\vdash \Gamma^\rho$ est prouvable dans $\mathsf{LL}^+_{\mathrm{pol}}$ mais $\Gamma = \Gamma^\rho$ puisque $\Gamma$ est sans atome. Soit $\pi$ une preuve sans coupure de $\vdash \Gamma$ dans $\mathsf{LL}^+_{\mathrm{pol}}$, par propriété de la sous-formule, elle n'utilise que des formules polarisées et c'est une preuve dans $\mathsf{LL}_{\mathrm{pol}}$. □

4.19. Lemme (Réciproque du lemme 4.3)
*Si $\vdash \Gamma, P, Q$ ne contenant que des formules polarisées est prouvable dans $\mathsf{LL}$ alors $\Gamma$ contient une formule négative avec $\top$ comme sous-formule sous la portée d'aucun connecteur exponentiel.*

Démonstration : Par récurrence sur la taille d'une preuve $\pi$ sans coupure de $\vdash \Gamma, P, Q$ en considérant chaque cas de dernière règle :

($ax$) Impossible puisque la conclusion de $\pi$ contient deux formules positives.

(1) Impossible puisque la conclusion de $\pi$ contient deux formules positives.

($\top$) Immédiat.

($\parr$, $\otimes$, $\&$, $\oplus_i$, $\perp$, $\forall$, $\exists$, $?d$, $?c$, $?w$) On applique l'hypothèse de récurrence à la (ou les) prémisse(s) ce qui donne le résultat, que la formule dans laquelle apparaît $\top$ soit active ou non.

($!$) Impossible puisque la conclusion de $\pi$ contient deux formules positives. □

4.20. Remarque : En particulier le séquent $\vdash P, Q$ ne peut pas être prouvable dans $\mathsf{LL}$.

4.21. Lemme
*Si le séquent $\vdash \mathcal{N}, P, Q$ est prouvable dans $\mathsf{LL}$ alors $\vdash \mathcal{N}, P, ?Q$ est prouvable dans $\mathsf{LL}_{\mathrm{pol}}$.*

Démonstration : Il s'agit de construire une preuve ne contenant pas de règles $\top$ introduisant plus d'une formule positive. D'après le lemme 4.19, $\mathcal{N} = \mathcal{N}', N$ où $N$ contient $\top$ sous la portée d'aucun connecteur exponentiel. On raisonne par récurrence sur la taille de $\mathcal{N}$ en considérant chaque cas de connecteur principal pour $N$ :

($\top$) Si $N = \top$ alors $\vdash \mathcal{N}', P, ?Q, \top$ est prouvable dans $\mathsf{LL}_{\mathrm{pol}}$.

($\parr$) Par le lemme 4.12, si $N = N_1 \parr N_2$ alors $\vdash \mathcal{N}', P, Q, N_1, N_2$ est prouvable dans $\mathsf{LL}$ d'où, par hypothèse de récurrence, $\vdash \mathcal{N}', P, ?Q, N_1, N_2$ est prouvable dans $\mathsf{LL}_{\mathrm{pol}}$ et enfin $\vdash \mathcal{N}', P, ?Q, N_1 \parr N_2$ également.

($\&$) Si $N = N_1 \& N_2$ alors, par le lemme 4.12, $\vdash \mathcal{N}', P, Q, N_1$ et $\vdash \mathcal{N}', P, Q, N_2$ sont prouvables dans $\mathsf{LL}$. Par hypothèse de récurrence, $\vdash \mathcal{N}', P, ?Q, N_1$ et $\vdash \mathcal{N}', P, ?Q, N_2$ sont prouvables dans $\mathsf{LL}_{\mathrm{pol}}$ donc $\vdash \mathcal{N}', P, ?Q, N_1 \& N_2$ également.

($\forall$) Si $N = \forall X N_1$ alors, par le lemme 4.12, $\vdash \mathcal{N}', P, Q, N_1$ est prouvable dans $\mathsf{LL}$ et, en appliquant l'hypothèse de récurrence, on obtient que $\vdash \mathcal{N}', P, ?Q, \forall X N_1$ est prouvable dans $\mathsf{LL}_{\mathrm{pol}}$. □

4.22. Proposition (Conservation propositionnelle)
*$\mathsf{LL}$ est une extension conservatrice de $\mathsf{LL}_{\mathrm{pol}}$ pour le fragment propositionnel. Ce qui signifie que, si $\vdash \Gamma$ est prouvable dans $\mathsf{LL}$ et ne contient que des formules polarisées sans quantificateur et qu'au plus une formule positive, alors $\vdash \Gamma$ est prouvable dans $\mathsf{LL}_{\mathrm{pol}}$.*

Démonstration : Soit $\pi$ une preuve sans coupure de $\vdash \Gamma$ dans LL, par propriété de la sous-formule, $\pi$ ne contient que des formules polarisées. Cependant, $\pi$ peut contenir des règles $\top$ introduisant plus d'une formule positive donc $\pi$ n'est pas nécessairement une preuve de $\mathsf{LL}_{\mathsf{pol}}$.

On raisonne par récurrence sur le nombre de séquents avec plus d'une formule positive apparaissant dans $\pi$. S'il n'y en a aucun, $\pi$ est une preuve de $\vdash \Gamma$ dans $\mathsf{LL}_{\mathsf{pol}}$. S'il y en a $n+1$, soit $\vdash \Delta$ un séquent contenant plus d'une formule positive et tel qu'aucun séquent situé au-dessous dans $\pi$ ne contienne plus d'une formule positive. Puisque la conclusion de $\pi$ ne contient qu'au plus une formule positive par hypothèse, il y a nécessairement une règle sous $\vdash \Delta$ et elle doit faire disparaître une formule positive, on en déduit que c'est une règle $?d$ et que $\vdash \Delta$ contient exactement deux formules positives donc $\Delta = \mathcal{N}, P, Q$ avec la règle $?d$ portant, par exemple, sur $Q$. En appliquant le lemme 4.21, on obtient que $\vdash \mathcal{N}, P, ?Q$ est prouvable dans $\mathsf{LL}_{\mathsf{pol}}$. Soit $\pi'$ la preuve obtenue en remplaçant la sous-preuve de $\pi$ qui démontre $\vdash \mathcal{N}, P, ?Q$ par sa preuve dans $\mathsf{LL}_{\mathsf{pol}}$, on peut appliquer l'hypothèse de récurrence à $\pi'$ et on en déduit que $\vdash \Gamma$ est prouvable dans $\mathsf{LL}_{\mathsf{pol}}$. □

4.23. Remarque : La propriété précédente est évidemment fausse si on considère de manière plus générale des séquents $\vdash \Gamma$ ne contenant que des formules polarisées (sans contrainte sur le nombre de formules positives) puisque, d'après le lemme 4.3, les séquents prouvables dans $\mathsf{LL}_{\mathsf{pol}}$ contiennent au plus une formule positive.

La proposition précédente et le lemme 4.18 permettent de montrer que les prouvabilités (pour les séquents propositionnels de LLP) dans LLP et dans LL sont équivalentes pour les formules sans atome. Autrement dit et avec un léger abus de langage, LL est une extension conservatrice de LLP pour le fragment propositionnel sans atome.

Question ouverte : LL est-il une extension conservatrice de $\mathsf{LL}_{\mathsf{pol}}$ au second ordre ?

La difficulté provient du fait que la règle $\exists$ de LL peut cacher des formules non polarisées, on ne peut donc plus appliquer simplement la propriété de la sous-formule. Un outil utile pour aborder cette question est probablement la *sémantique des phases* [Gir87] mais son utilisation devient complexe au second ordre [Oka99].

4.24. Exemple

Par contre LL n'est pas une extension conservatrice de LLP au second ordre puisque le séquent $\vdash X \otimes X, X^\perp$ n'est pas prouvable dans LL alors qu'il l'est dans LLP :

$$\dfrac{\dfrac{\dfrac{}{\vdash X, X^\perp}\,ax \qquad \dfrac{}{\vdash X, X^\perp}\,ax}{\vdash X \otimes X, X^\perp, X^\perp}\,\otimes}{\vdash X \otimes X, X^\perp}\,?c$$

## 4.3 Traductions vers des fragments

Afin de clarifier les résultats de cette section, on ne considère que la partie propositionnelle de LLP. L'extension aux quantificateurs ne pose pas de problème (remarque 4.34) mais ne sera pas utile dans la suite.

On va montrer que, sous certaines conditions, on peut coder LLP dans certains de ses fragments modulo des isomorphismes. Ceci permet de ramener, pour certaines questions, l'étude de LLP à l'étude de fragments plus simples et plus appropriés.

4.25. Définition (Isomorphisme syntaxique)
Soient $A$ et $B$ deux formules, la paire $(\pi_A, \pi_B)$ est un *isomorphisme syntaxique* entre $A$ et $B$ si :
- $\pi_A$ est une preuve de $\vdash A, B^\perp$.
- $\pi_B$ est une preuve de $\vdash A^\perp, B$.
- La preuve obtenue en coupant les conclusions $A$ et $A^\perp$ de $\pi_A$ et $\pi_B$ se normalise en une $\eta$-expansion de l'axiome $\vdash B^\perp, B$.
- La preuve obtenue en coupant les conclusions $B^\perp$ et $B$ de $\pi_A$ et $\pi_B$ se normalise en une $\eta$-expansion de l'axiome $\vdash A, A^\perp$.

4.26. Lemme (Polarités isomorphes)
*Soient $A$ et $B$ deux formules polarisées syntaxiquement isomorphes, $A$ et $B$ ont la même polarité.*

Démonstration : Raisonnons par l'absurde, en supposant par exemple que $A$ est positive et $B$ est négative alors, tout simplement, $\vdash A, B^\perp$ n'est pas prouvable par la remarque 4.20. □

Les traductions internes de LLP dans des fragments sont toutes basées sur l'utilisation des isomorphismes suivants :

4.27. Proposition (Quelques isomorphismes syntaxiques polarisés)
*Les équivalences suivantes sont réalisées par des isomorphismes syntaxiques :*

$$A \parr (B \mathbin{\&} C) \multimapboth (A \parr B) \mathbin{\&} (A \parr C) \tag{4.1}$$
$$A \otimes (B \oplus C) \multimapboth (A \otimes B) \oplus (A \otimes C) \tag{4.2}$$
$$A \parr \top \multimapboth \top \tag{4.3}$$
$$A \otimes 0 \multimapboth 0 \tag{4.4}$$
$$!(A \mathbin{\&} B) \multimapboth\ !A \otimes !B \tag{4.5}$$
$$?(A \oplus B) \multimapboth\ ?A \parr ?B \tag{4.6}$$
$$!\top \multimapboth 1 \tag{4.7}$$
$$?0 \multimapboth \perp \tag{4.8}$$

Démonstration : Les équations 4.1 et 4.2 sont en fait les mêmes par dualité, on n'a donc que deux implications à démontrer :

$$\begin{prooftree}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash A^\perp, A$}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash B^\perp, B$}
\RightLabel{$\oplus_1$}
\UnaryInfC{$\vdash B^\perp \oplus C^\perp, B$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash A^\perp \otimes (B^\perp \oplus C^\perp), A, B$}
\RightLabel{$\parr$}
\UnaryInfC{$\vdash A^\perp \otimes (B^\perp \oplus C^\perp), A \parr B$}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash A^\perp, A$}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash C^\perp, C$}
\RightLabel{$\oplus_2$}
\UnaryInfC{$\vdash B^\perp \oplus C^\perp, C$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash A^\perp \otimes (B^\perp \oplus C^\perp), A, C$}
\RightLabel{$\parr$}
\UnaryInfC{$\vdash A^\perp \otimes (B^\perp \oplus C^\perp), A \parr C$}
\RightLabel{$\&$}
\BinaryInfC{$\vdash A^\perp \otimes (B^\perp \oplus C^\perp), (A \parr B) \mathbin{\&} (A \parr C)$}
\end{prooftree}$$

$$\begin{prooftree}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash A, A^\perp$}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash B, B^\perp$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash A, B, A^\perp \otimes B^\perp$}
\RightLabel{$\oplus_1$}
\UnaryInfC{$\vdash A, B, (A^\perp \otimes B^\perp) \oplus (A^\perp \otimes C^\perp)$}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash A, A^\perp$}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash C, C^\perp$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash A, C, A^\perp \otimes C^\perp$}
\RightLabel{$\oplus_2$}
\UnaryInfC{$\vdash A, C, (A^\perp \otimes B^\perp) \oplus (A^\perp \otimes C^\perp)$}
\RightLabel{$\&$}
\BinaryInfC{$\vdash A, B \mathbin{\&} C, (A^\perp \otimes B^\perp) \oplus (A^\perp \otimes C^\perp)$}
\RightLabel{$\parr$}
\UnaryInfC{$\vdash A \parr (B \mathbin{\&} C), (A^\perp \otimes B^\perp) \oplus (A^\perp \otimes C^\perp)$}
\end{prooftree}$$

De même pour les équations 4.3 et 4.4 :

$$\dfrac{\dfrac{}{\vdash A, \top, 0}\;\top}{\vdash A \parr \top, 0}\;\parr \qquad\qquad \dfrac{}{\vdash A^{\perp} \otimes 0, \top}\;\top$$

Et de même pour les équations 4.5 et 4.6 :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A^{\perp}, A}\;ax}{\vdash A^{\perp} \oplus B^{\perp}, A}\;\oplus_1}{\vdash ?(A^{\perp} \oplus B^{\perp}), A}\;?d}{\vdash ?(A^{\perp} \oplus B^{\perp}), !A}\;! \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\vdash B^{\perp}, B}\;ax}{\vdash A^{\perp} \oplus B^{\perp}, B}\;\oplus_2}{\vdash ?(A^{\perp} \oplus B^{\perp}), B}\;?d}{\vdash ?(A^{\perp} \oplus B^{\perp}), !B}\;!}{\vdash ?(A^{\perp} \oplus B^{\perp}), ?(A^{\perp} \oplus B^{\perp}), !A \otimes !B}\;\otimes}{\vdash ?(A^{\perp} \oplus B^{\perp}), !A \otimes !B}\;?c$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A, A^{\perp}}\;ax}{\vdash A, ?A^{\perp}}\;?d}{\vdash A, ?A^{\perp}, ?B^{\perp}}\;?w \qquad \dfrac{\dfrac{\dfrac{}{\vdash B, B^{\perp}}\;ax}{\vdash B, ?B^{\perp}}\;?d}{\vdash B, ?A^{\perp}, ?B^{\perp}}\;?w}{\vdash A \,\&\, B, ?A^{\perp}, ?B^{\perp}}\;\&}{\vdash !(A \,\&\, B), ?A^{\perp}, ?B^{\perp}}\;!}{\vdash !(A \,\&\, B), ?A^{\perp} \parr ?B^{\perp}}\;\parr$$

Et encore pour les équations 4.7 et 4.8 :

$$\dfrac{\dfrac{\dfrac{}{\vdash \top}\;\top}{\vdash !\top}\;!}{\vdash !\top, \perp}\;\perp \qquad\qquad \dfrac{\dfrac{}{\vdash 1}\;1}{\vdash ?0, 1}\;?w$$

□

Les propositions suivantes sont basées sur l'application des équations 4.1, 4.2, 4.3 et 4.4 de la gauche vers la droite et des autres équations dans la direction appropriée.

4.28. Proposition (Traduction interne additive)
*Soit $A$ une formule polarisée* sans atome*, il existe une formule polarisée sans atome $A'$ syntaxiquement isomorphe à $A$ et ne contenant aucun connecteur multiplicatif.*

Démonstration : On va appliquer les équations 4.5, 4.6, 4.7, et 4.8 de la droite vers la gauche.

On raisonne par récurrence sur le couple (nombre de connecteurs multiplicatifs dans $A$, taille de $A$). S'il n'y a pas de connecteur multiplicatif dans $A$, le résultat est prouvé en prenant $A' = A$. S'il y en a, on considère chaque cas de connecteur principal de $A$ sachant que $A$ n'est pas un atome. Considérons le cas où $A$ est négative, le cas positif s'obtenant directement par dualité :

($\perp$) On a $\perp$ syntaxiquement isomorphe à $?0$ par l'équation 4.8.

($\top$) Pas de connecteur multiplicatif.

($\parr$) On a $A = B \parr C$, l'hypothèse de récurrence donne deux formules sans connecteur multiplicatif $B'$ et $C'$ syntaxiquement isomorphes à $B$ et $C$. Par polarisation, $B$ et $C$ sont négatives donc $B'$ et $C'$ aussi (lemme 4.26). Puisqu'elles ne contiennent pas de connecteur multiplicatif, leur connecteur principal ne peut être que $\top$, & ou ?.
- Si l'une a pour connecteur principal $\top$ alors $A$ est syntaxiquement isomorphe à $\top$ par l'équation 4.3.

- Si l'une a pour connecteur principal &, supposons que c'est $B' = B_1 \,\&\, B_2$ alors on applique l'hypothèse de récurrence à $B_1 \parr C$ et $B_2 \parr C$, ce que l'on peut faire puisque soit $B$ contient un connecteur multiplicatif et alors $B_i \parr C$ en contient moins que $A$ soit $B$ n'en contient pas et alors on a déjà vu que $B' = B$ donc $B_i \parr C$ est plus petite que $A$. Ainsi on obtient deux formules $B'_1$ et $B'_2$ et il suffit de considérer $A' = B'_1 \,\&\, B'_2$ par l'équation 4.1.
- Sinon les deux ont pour connecteur principal ? donc $B' = ?B_1$ et $C' = ?C_1$. On a alors $A' = ?(B_1 \oplus C_1)$ syntaxiquement isomorphe à $A$ par l'équation 4.6.

(&) On applique l'hypothèse de récurrence aux sous-formules immédiates.

(?) On applique l'hypothèse de récurrence à la sous-formule immédiate. □

On trouvera une manière un petit peu différente de prouver ce résultat à travers le lemme 7.29.

4.29. Lemme
*Soit $A$ une formule polarisée ne commençant pas par un connecteur exponentiel, il existe une formule polarisée $A'$ syntaxiquement isomorphe à $A$ telle que soit le connecteur principal de $A'$ est un connecteur additif, soit $A'$ ne contient aucun connecteur additif.*

Démonstration : On applique les équations 4.5, 4.6, 4.7 et 4.8 de la gauche vers la droite.

On suppose que $A$ est négative, le cas positif s'obtient immédiatement par dualité. On raisonne par récurrence sur le couple (nombre de connecteurs additifs dans $A$, taille de $A$) en supposant de plus que si $A$ ne contient pas de connecteur additif alors $A' = A$. On considère chaque cas de connecteur principal pour $A$ :

($\bot$) Pas de connecteur additif donc $A' = A = \bot$.

($\top$) $A' = A = \top$ a un connecteur principal additif.

($\parr$) Si $A = B \parr C$, on applique l'hypothèse de récurrence à $B$ et $C$ ce qui donne deux formules $B'$ et $C'$, on considère les cas suivants :
- Si le connecteur principal de $B'$ ou de $C'$ est $\top$ alors $A' = \top$ est syntaxiquement isomorphe à $A$ par l'équation 4.3.
- Si $B' = B_1 \,\&\, B_2$, alors $A' = (B_1 \parr C') \,\&\, (B_2 \parr C')$ est syntaxiquement isomorphe à $A$ par l'équation 4.1 et a un connecteur principal additif.
- De même si le connecteur principal de $C'$ est &.
- Sinon, $B'$ et $C'$ sont sans connecteur additif et on prend $A' = B' \parr C'$.

(&) $A' = A = B \,\&\, C$ a un connecteur principal additif. □

4.30. Proposition (Traduction interne multiplicative)
*Soit $A$ une formule polarisée commençant par un connecteur exponentiel, il existe une formule polarisée $A'$ syntaxiquement isomorphe à $A$ et ne contenant aucun connecteur additif.*

Démonstration : On applique les équations 4.5, 4.6, 4.7 et 4.8 de la gauche vers la droite.

On suppose que $A$ est positive, donc par hypothèse $A = !N$, le cas négatif s'obtient immédiatement par dualité et on raisonne par récurrence sur la taille de $A$. Si $N$ commence elle-même par un connecteur exponentiel, on lui applique l'hypothèse de récurrence. Sinon, on applique le lemme 4.29 à $N$ ce qui nous donne une formule $N'$ syntaxiquement isomorphe à $N$. Si $N'$ ne contient pas de connecteur additif, on prend $A' = !N'$. Sinon, le connecteur principal de $N'$ est un connecteur additif :

($\top$) On a $!\top$ syntaxiquement isomorphe à 1 par l'équation 4.7, donc on prend $A' = 1$.

(&) Si $N' = B\&C$, on applique l'hypothèse de récurrence à $!B$ et $!C$ ce qui donne les formules $B'$ et $C'$. On a $A' = B' \otimes C'$ syntaxiquement isomorphe à $A$ par l'équation 4.5. □

4.31. Remarque : En appliquant la même méthode que pour la traduction multiplicative, on peut facilement montrer que toute formule polarisée est soit syntaxiquement équivalente à une formule polarisée sans $\top$ soit syntaxiquement équivalente à $\top$ puisque :

$$\begin{aligned} \top \parr A &\multimap\!\!\circ\, \top \\ \top \,\&\, A &\multimap\!\!\circ\, A \\ !\top &\multimap\!\!\circ\, 1 \end{aligned}$$

4.32. Proposition (Traductions internes positive et négative)
*Soit $A$ une formule positive (resp. négative) sans atome, il existe une formule polarisée $A'$ syntaxiquement isomorphe à $A$ et ne contenant aucun connecteur négatif (resp. positif).*

Démonstration : Comme pour les deux propositions précédentes. □

4.33. Exemple
En distribuant $\parr$ sur & et $\otimes$ sur $\oplus$, la formule polarisée $!(?1 \parr (\bot \,\&\, ?(!\top \otimes (1 \oplus !\bot))))$ est syntaxiquement isomorphe à $!(?1 \,\&\, (?1 \parr ?(!\top \oplus (!\top \otimes !\bot))))$, et par les traductions décrites dans les propositions précédentes, on obtient les formules syntaxiquement isomorphes :
- $!(?!\top \,\&\, ?(!\top \oplus (!\top \oplus !?0)))$ sans connecteur multiplicatif,
- $!?1 \otimes !(?1 \parr (?1 \parr ?!\bot))$ sans connecteur additif,
- $!?1 \otimes !?(1 \oplus (1 \oplus !?0))$ sans connecteur négatif.

4.34. Remarque : Si on ajoute à la proposition 4.27, les isomorphismes syntaxiques suivants :

$$\begin{aligned} A \parr (\forall X B) &\multimap\!\!\circ\, \forall X(A \parr B) && X \notin A \\ A \otimes (\exists X B) &\multimap\!\!\circ\, \exists X(A \otimes B) && X \notin A \\ \forall X(A \,\&\, B) &\multimap\!\!\circ\, (\forall X A) \,\&\, (\forall X B) \\ \exists X(A \oplus B) &\multimap\!\!\circ\, (\exists X A) \oplus (\exists X B) \\ \forall X \top &\multimap\!\!\circ\, \top \\ \exists X 0 &\multimap\!\!\circ\, 0 \end{aligned}$$

on peut étendre les résultats précédents à LLP avec quantificateurs en appliquant ces isomorphismes de la gauche vers la droite en même temps que 4.1 et 4.2 au début de chaque traduction.

# Chapitre 5

# Réseaux de preuve polarisés

Les réseaux de preuve constituent la principale nouveauté syntaxique due à LL. Ils permettent de simplifier grandement l'étude de l'élimination des coupures en s'affranchissant de la séquentialité parfois trop importante du calcul des séquents. Nous verrons que la contrainte de polarisation permet d'une part de simplifier les critères de correction (grâce à l'introduction d'une nouvelle orientation des réseaux) et d'autre part de définir une élimination des coupures complète pour tout le système LLP. La principale difficulté provient des règles structurelles généralisées et des connecteurs additifs. Nous aborderons trois solutions possibles pour l'introduction de ces connecteurs : les réseaux à boîtes, les réseaux à poids et les réseaux par tranches, ce qui permet une présentation de moins en moins séquentielle de ces objets.

Une grande partie des résultats de ce chapitre proviennent de [Lau99, Lau02b, LQTdF00, LTdF01].

## 5.1 Réseaux à boîtes additives

### 5.1.1 Structures de preuve à boîtes

Nous allons définir la notion de *structure de preuve* pour LLP. Par rapport à la section 1.1, nous allons ajouter les contraintes propres à LLP mais également étendre le langage aux connecteurs additifs et aux constantes à partir de ce qui a été fait dans [Gir87, Reg92].

5.1. Définition (Structure de preuve polarisée à boîtes additives)
Une *structure de preuve polarisée à boîtes additives* est une structure de preuve au sens de la définition 1.1 construite sur les nœuds de la figure 5.1 telle que les types des arêtes soient des formules polarisées et concernant les boîtes :

- Les boîtes des nœuds ! ont pour conclusions $\mathcal{N}, N$.
- À chaque nœud & de prémisses $N$ et $M$ sont associées deux sous-structures de preuve (appelées boîtes) de conclusions $\Gamma, N$ et $\Gamma, M$ telles que si $A \in \Gamma$, les deux conclusions des boîtes correspondant à $A$ sont prémisses du même nœud C.
- Tout nœud C a pour prémisses les conclusions de deux boîtes associées au même nœud &.

Ce nouveau nœud C sert à traduire l'opération de contraction sur le contexte qui a lieu dans la règle &.

5.2. Définition (Arêtes et nœuds positifs et négatifs)
Une arête est *positive* (resp. *négative*) si son type est une formule positive (resp. négative).

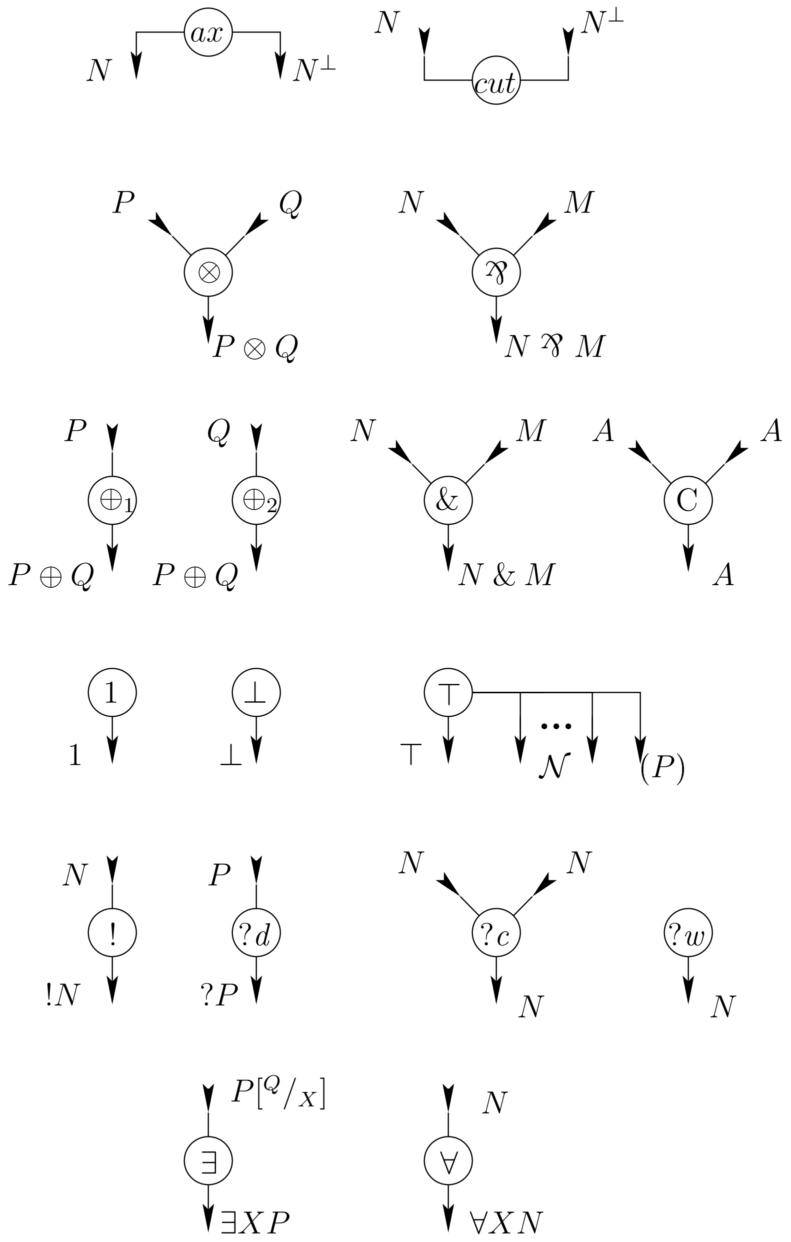



FIG. 5.1 – Nœuds des structures de preuve polarisées à boîtes additives avec la *s-orientation* (le nœud $\top$ peut ne pas avoir de conclusion positive).

Un nœud est *positif* (resp. *négatif*) si toutes ses arêtes sont positives (resp. négatives). Les nœuds positifs sont $\otimes$, $\oplus_i$, 1, $\exists$ et les nœuds C dont le type est positif (voir section 5.1.3 à ce sujet). Les nœuds négatifs sont $⅋$, &, $\bot$, $\top$ (lorsqu'il n'a pas de conclusion positive voir section 5.1.3), $?c$, $?w$, $\forall$ et les nœuds C dont le type est négatif.

5.3. Définition (Traduction du calcul des séquents)
La traduction des règles multiplicatives et exponentielles est la même que dans la définition 1.3, on y ajoute celle des autres règles. Soit $\pi$ une preuve de $\vdash \Gamma$ en calcul des séquents, on lui associe la structure de preuve $\mathcal{S}_\pi$ de conclusions $\Gamma$, définie récursivement à partir de $\pi$ en regardant la dernière règle $R$ (on note $\pi_1$ ou $\pi_1$ et $\pi_2$ les preuves des prémisses de $R$ selon son arité) :

(&) On ajoute à $\mathcal{S}_{\pi_1}$ et $\mathcal{S}_{\pi_2}$ des liens C entre les conclusions correspondant deux à deux aux formules du contexte et on ajoute un lien & entre les conclusions correspondant aux formules actives. Les boîtes du nouveau lien & sont $\mathcal{S}_{\pi_1}$ et $\mathcal{S}_{\pi_2}$.

($\oplus_i$) On ajoute à $\mathcal{S}_{\pi_1}$ un lien $\oplus_i$ sous la conclusion correspondant à la formule active.

(1) $\mathcal{S}_\pi$ contient un unique lien 1.

($\bot$) On ajoute à $\mathcal{S}_{\pi_1}$ un lien $\bot$.

($\top$) $\mathcal{S}_\pi$ contient un unique lien $\top$ avec comme conclusions : $\top$ et le contexte de la règle.

On pourrait étendre la définition de structure de preuve *séquentialisable* aux connecteurs que l'on vient d'ajouter et redémontrer le lemme 1.8. On se contentera d'assimiler structure de preuve séquentialisable et traduction du calcul des séquents.

### 5.1.2 Critère de correction

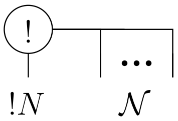
5.4. Définition (Orientations)
La définition de structure de preuve donne une orientation des arêtes que l'on appelle *orientation structurelle* ou *s-orientation*. Comme expliqué à la section 1.1, on y fait référence par les termes *au-dessus*, *au-dessous*, *haut*, *bas*, *monter*, *descendre*, ...

On définit une deuxième orientation sur les structures de preuve polarisées, appelée *orientation polarisée* ou *p-orientation*, en orientant les arêtes négatives *vers le bas* (donc dans le même sens que pour la s-orientation) et les arêtes positives *vers le haut* (donc dans le sens opposé à la s-orientation).

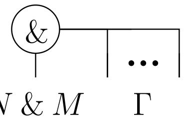
Excepté dans le cas des termes ci-dessus et sauf mention explicite, on considérera désormais les structures de preuve polarisées comme des graphes orientés avec la p-orientation (en particulier dans les définitions qui suivent). On fera référence à la p-orientation par les termes *entrant*, *sortant*, *initial*, *final*, ...

5.5. Définition (Graphe de correction)
Le *graphe de correction* à profondeur 0 d'une structure de preuve polarisée à boîtes additives $\mathcal{S}$ est le graphe orienté par la *p-orientation* contenant les nœuds à profondeur 0 de $\mathcal{S}$ et dans lequel :

- On supprime les arêtes pendantes.
- Chaque lien ! dont la boîte a pour conclusions $\mathcal{N}, N$ est remplacé par un lien ! généralisé sans prémisse et avec comme conclusions $\mathcal{N}, !N$ :




- Chaque lien & dont les boîtes ont pour conclusions $\Gamma, N$ et $\Gamma, M$, ainsi que les nœuds C associés, est remplacé par un lien & généralisé sans prémisse et avec comme conclusions $\Gamma, N \,\&\, M$ :




– Si $n$ est un lien $\forall$, pour chaque lien $n'$ qui *dépend* de la *variable propre* de $n$ on ajoute une arête orientée de $n'$ vers $n$, appelée *arête de saut*.

Un graphe de correction de $\mathcal{S}$ est soit son graphe de correction à profondeur 0 soit le graphe de correction à profondeur 0 d'une des boîtes de $\mathcal{S}$.

5.6. Remarque : Le point important ici est qu'en utilisant la p-orientation, on se contente de considérer un seul graphe de correction à profondeur 0, et il n'est pas nécessaire de mutiler les nœuds $⅋$ et $?c$.

Par conséquent, un nœud d'une structure de preuve apparaît dans exactement un de ses graphes de correction. En particulier, on va obtenir un critère de correction dont la linéarité est immédiate alors que ce résultat est nettement plus difficile pour MELL [Gue99].

5.7. Remarque : Un graphe de correction ne contient pas de nœud C.

5.8. Définition (Nœuds initiaux et finaux)
Un *nœud initial* (resp. *final*) d'une structure de preuve est un nœud sans arête entrante (resp. sortante) dans le graphe de correction auquel il appartient.

5.9. Remarques :
– Les nœuds $?d$, $?w$ et $\perp$ sont toujours initiaux, les nœuds 1 sont toujours finaux.
– Un nœud terminal ayant une conclusion positive est toujours initial (car aucun nœud à conclusion positive n'a de prémisse négative ou d'autre conclusion positive dans un graphe de correction).
– Un nœud négatif terminal est toujours final (la seule difficulté pourrait provenir d'une arête de saut partant de ce nœud, or il n'y a pas de variables propres dans les conclusions d'une structure de preuve).

5.10. Définition (Réseau de preuve polarisé)
Une structure de preuve polarisée à boîtes additives est *correcte* ou est un *réseau de preuve polarisé à boîtes additives* (ou simplement un *réseau polarisé*) si tous ses graphes de correction sont acycliques et possèdent exactement un nœud initial non affaiblissement (*i.e.* ni $?w$ ni $\perp$).

5.11. Remarque : Le nœud initial non affaiblissement est soit un nœud $?d$ soit un nœud positif terminal soit un nœud ! ou & généralisé terminal soit un nœud $\top$ terminal.

5.12. Définition (Royaume)
Le *royaume* d'une arête $a$ dans un réseau $\mathcal{R}$ est le plus petit sous-réseau de $\mathcal{R}$ contenant $a$.

5.13. Définition (Arbre positif)
On définit récursivement la sous-structure de preuve $\mathcal{A}$ contenant l'ensemble des nœuds situés au-dessus d'une arête positive $a$, appelé *arbre positif* de l'arête, selon le nœud dont $a$ est conclusion :

($ax$) $\mathcal{A}$ est réduit au nœud $ax$.

(!) $\mathcal{A}$ est constitué du nœud ! et de sa boîte.

(C) $\mathcal{A}$ est constitué du nœud & correspondant, de ses boîtes et de tous les nœuds C associés.

(1) $\mathcal{A}$ est réduit au nœud 1.

($\top$) $\mathcal{A}$ est réduit au nœud $\top$.

($\otimes$) Soient $\mathcal{A}_1$ et $\mathcal{A}_2$ les arbres positifs des prémisses du nœud $\otimes$, $\mathcal{A}$ contient $\mathcal{A}_1$, $\mathcal{A}_2$ et le nœud $\otimes$.

($\oplus_i$) Soit $\mathcal{A}_1$ l'arbre positif de la prémisse du nœud $\oplus_i$, $\mathcal{A}$ contient $\mathcal{A}_1$ et le nœud $\oplus_i$.

($\exists$) Soit $\mathcal{A}_1$ l'arbre positif de la prémisse du nœud $\exists$, $\mathcal{A}$ contient $\mathcal{A}_1$ et le nœud $\exists$.

L'arête $a$ est appelée *racine* de l'arbre. Les arêtes conclusion des nœuds *ax*, &, C, $\top$ (et les portes auxiliaires de boîtes !) qui sont dans $\mathcal{A}$ mais ne sont pas héréditairement au-dessus de $a$ sont appelées *feuilles* de l'arbre, ce sont des arêtes négatives.

5.14. Lemme (Royaume positif)
*Le royaume d'une arête positive est son arbre positif.*

Démonstration : Soit $a$ une arête positive d'un réseau $\mathcal{R}$, le royaume de $a$ dans $\mathcal{R}$ est une sous-structure de preuve, elle doit donc contenir pour chaque arête le nœud dont elle est conclusion et pour chaque nœud ! la boîte qui lui est associée (et de même pour les nœuds &). On en déduit que le royaume de $a$ contient l'arbre positif de $a$. Il reste à montrer que c'est un réseau, ce qui est immédiat récursivement à partir de la définition 5.13. □

5.15. Lemme (Ordre partiel)
*On définit, sur les nœuds et les arêtes d'un réseau de preuve polarisé, la relation $x \leq y$ si $x$ et $y$ appartiennent au même graphe de correction et s'il existe un chemin p-orienté de $x$ à $y$ dans ce graphe ($x$ et $y$ pouvant être deux nœuds, deux arêtes ou un nœud et une arête).*

*La relation $\leq$ est une relation d'ordre partiel.*

Démonstration : La seule propriété non triviale est l'anti-symétrie qui découle de l'acyclicité des graphes de correction. □

5.16. Remarques :
- Les nœuds initiaux sont les nœuds minimaux pour l'ordre $\leq$, tout nœud est donc accessible à partir d'un nœud initial selon un chemin p-orienté.
- Si $a$ est une arête positive, pour toute arête $b$ de l'arbre de $a$ à même profondeur que $a$, on a $a \leq b$.

La présence de boîtes pour les connecteurs additifs nous impose une gestion un peu particulière de & et $\top$ par rapport aux autres connecteurs négatifs. On verra dans les sections 5.2 et 5.3 que ceci ne cache rien de significatif.

5.17. Lemme (Descente négative)
*Soit $a$ une arête négative à profondeur 0 d'une structure de preuve polarisée, toutes les arêtes situées sous $a$ sont négatives, ou autrement dit l'une des trois propriétés suivantes est vérifiée :*

1. *$a$ est une conclusion du réseau.*
2. *$a$ est située héréditairement au-dessus d'une coupure.*
3. *$a$ est située héréditairement au-dessus d'un nœud négatif terminal non additif.*

Démonstration : Puisque $a$ est à profondeur 0 et puisque tout lien négatif additif (C, & ou $\top$) a ses prémisses dans une boîte (ou n'a pas de prémisses), un tel nœud ne peut pas se trouver sous $a$.

On raisonne par récurrence sur le nombre de nœuds situés héréditairement au-dessous de $a$ :
- S'il n'y en a aucun, $a$ est conclusion du réseau : cas 1.
- S'il y en a 1, le nœud situé sous $a$ peut être soit une coupure (et on est dans le cas 2) soit un nœud $\parr$, $?c$ ou $\forall$ qui est un nœud négatif et terminal (cas 3).

– S'il y en a $n > 1$, le nœud situé sous $a$ est un nœud $⅋$, $?c$ ou $\forall$ qui possède une conclusion négative à laquelle on peut appliquer l'hypothèse de récurrence. □

5.18. Lemme (Descente positive)
*Soit $a$ une arête positive à profondeur 0 d'une structure de preuve polarisée, l'une des trois propriétés suivantes est vérifiée :*

1. *$a$ est située héréditairement au-dessus d'une conclusion positive du réseau.*
2. *$a$ est située héréditairement au-dessus d'une coupure.*
3. *$a$ est située héréditairement au-dessus d'un nœud $?d$.*

Démonstration : Par récurrence sur le nombre de nœuds situés héréditairement au-dessous de $a$ :
– S'il n'y en a aucun, $a$ est conclusion positive du réseau : cas 1.
– S'il y en a $n > 0$, le nœud situé sous $a$ peut être soit une coupure (et on est dans le cas 2), soit un nœud $?d$ (cas 3) soit un nœud $\otimes$, $\oplus_i$ ou $\exists$ qui possède une conclusion positive à laquelle on peut appliquer l'hypothèse de récurrence. □

5.19. Théorème (Séquentialisation)
*Une structure de preuve est séquentialisable* ssi *elle est correcte.*

Démonstration : La première implication est immédiate par récurrence sur la taille de la structure de preuve.

Nous démontrons donc la deuxième implication. Soit $\mathcal{R}$ un réseau de preuve polarisé (ou une structure de preuve polarisée correcte), on note $\mathcal{G}$ son graphe de correction à profondeur 0. La preuve se fait par récurrence sur la taille de $\mathcal{R}$ en considérant trois cas :
– Si $\mathcal{G}$ possède un nœud négatif terminal non additif, on considère chaque cas de tel nœud :
 ($⅋$, $?c$) En supprimant ce nœud, on obtient une structure de preuve $\mathcal{R}'$ dont on voit aisément qu'elle est correcte donc séquentialisable par hypothèse de récurrence. On en déduit que $\mathcal{R}$ est séquentialisable.
 ($\bot$, $?w$) En supprimant ce nœud, on obtient un réseau $\mathcal{R}'$ qui est séquentialisable par hypothèse de récurrence puisqu'on n'a touché qu'à des nœuds initiaux affaiblissements.
 ($\forall$) En supprimant ce nœud $n$, on obtient un réseau $\mathcal{R}'$ qui est séquentialisable par hypothèse de récurrence. On en déduit que $\mathcal{R}$ est séquentialisable puisque les conclusions de $\mathcal{R}'$ (excepté la prémisse de $n$) ne contiennent pas la variable propre de $n$ libre et la règle de $\forall$ peut donc s'appliquer.
– Si $\mathcal{G}$ ne possède aucun nœud négatif terminal non additif mais contient des nœuds *cut*, on considère un nœud *cut* $n$ maximal pour l'ordre $\leq$ parmi les nœuds *cut* de $\mathcal{G}$. Soit $a$ (resp. $a'$) la prémisse positive (resp. négative) de $n$ et soit $b$ une feuille de l'arbre positif $\mathcal{A}$ de $a$, on a $a \leq b$. Le lemme 5.17 appliqué à $b$ implique que $b$ est conclusion de $\mathcal{R}$ car $\mathcal{G}$ n'a pas de nœud négatif terminal non additif et si $b$ était situé au-dessus d'un nœud *cut* $n'$, on aurait $n \leq a \leq b \leq n'$ d'où $n = n'$ par maximalité de $n$ et donc on aurait un cycle dans $\mathcal{G}$.
 Soit $\mathcal{R}'$ la sous-structure de preuve de $\mathcal{R}$ contenant tous les nœuds exceptés $n$ et ceux de $\mathcal{A}$, on va montrer que $\mathcal{R}'$ est un réseau. La première chose est qu'il s'agit d'une structure de preuve, la difficulté vient de la contrainte sur l'absence de variable propre en conclusion. Pour cela, il faut montrer que le type de $a'$ ne contient pas de variable propre libre. Raisonnons par l'absurde, si c'était le cas, le type de $a$ contiendrait également cette variable et on aurait donc dans $\mathcal{G}$ une arête du lien prémisse de $a$ vers le lien $\forall$ correspondant à cette variable. Ce lien n'est pas terminal par hypothèse et on note $c$ sa conclusion. Par le lemme 5.17,

soit $c$ est au-dessus d'un nœud négatif terminal non additif ce qui est impossible, soit $c$ est au-dessus d'une coupure. Si cette coupure $n'$ n'est pas $n$ ceci contredit la maximalité de $n$ puisqu'on a un chemin de $n$ vers $c$ puis vers $n'$ et si $n' = n$ on a un cycle dans $\mathcal{G}$.
Montrer alors que $\mathcal{R}'$ est un réseau est immédiat. $\mathcal{A}$ étant également un réseau (lemme 5.14), on en conclut en appliquant l'hypothèse de récurrence à $\mathcal{R}'$ et à $\mathcal{A}$ que $\mathcal{R}$ est séquentialisable.

- Si $\mathcal{G}$ ne possède ni nœud négatif terminal non additif ni nœud *cut*. Si $\mathcal{R}$ possède une arête positive à profondeur 0, on applique le lemme 5.18 à cette arête et on en déduit que :
  - Soit $\mathcal{R}$ possède un nœud $?d$ qui est terminal en appliquant le lemme 5.17 à sa conclusion. Et on applique l'hypothèse de récurrence au réseau obtenu en supprimant ce nœud.
  - Soit $\mathcal{R}$ possède une conclusion positive $a$ et $\mathcal{R}$ est l'arbre positif $\mathcal{A}$ de $a$ (donc séquentialisable par la définition 5.13).
    Sinon on considère une arête $b$ qui ne soit pas dans $\mathcal{A}$, si elle est négative on a une coupure ou un nœud terminal non additif dans $\mathcal{R}$ par le lemme 5.17 et si elle est positive, par le lemme 5.18, il y a un nœud $?d$ ou une conclusion positive autre que $a$ (puisque $b \notin \mathcal{A}$) dans $\mathcal{R}$ et donc au moins deux sommets initiaux non affaiblissements (d'après la remarque 5.9) ; $b$ ne peut donc pas exister.

  Si $\mathcal{R}$ ne possède que des arêtes négatives à profondeur 0 alors, par le lemme 5.17 et puisque les nœuds négatifs additifs sont les seuls nœuds négatifs à n'avoir pas de prémisses dans $\mathcal{G}$, $\mathcal{G}$ ne possède que des nœuds $\top$ et &. De plus ces nœuds sont initiaux (pas d'arêtes positives) donc il n'y en a qu'un. On considère chacun des cas :
  (&) On applique l'hypothèse de récurrence à chacune des deux boîtes du nœud.
  ($\top$) Une structure de preuve réduite à un unique nœud $\top$ est séquentialisable. □

5.20. Remarque : Contrairement à la plupart des preuves de séquentialisation, nous avons ici une preuve «par le bas» : on utilise directement une procédure qui fournit une dernière règle.

5.21. Remarque : On a pu donner un critère de correction en présence d'affaiblissements et de constantes multiplicatives. L'exemple 1.12 ne contredit pas ce résultat car il n'est pas polarisable, de même que la version simplifiée $(1 \parr 1) \otimes \bot$ de cet exemple.

5.22. Corollaire (Correction sans coupure)
*Une structure de preuve sans coupure, qui possède à profondeur* 0 *et dans chaque boîte un unique nœud initial (non affaiblissement), est correcte.*

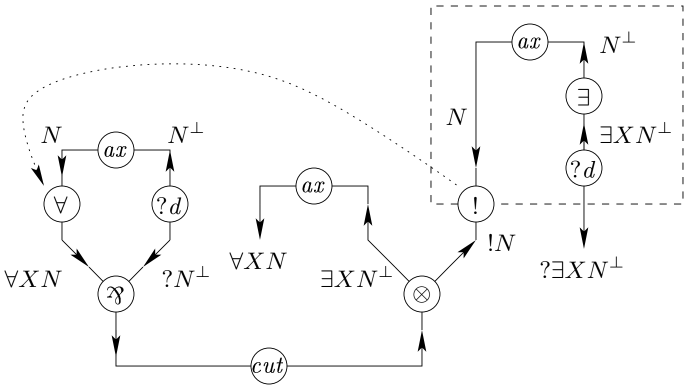
Démonstration : Par le théorème de séquentialisation et en raisonnant par récurrence sur la profondeur de la structure de preuve $\mathcal{S}$, il suffit de montrer que le graphe de correction $\mathcal{G}$ à profondeur 0 de $\mathcal{S}$ est acyclique. On raisonne par l'absurde en supposant que $\mathcal{G}$ contient un cycle $\sigma$. On va montrer, par récurrence sur le nombre d'arêtes de saut que contient ce cycle, qu'il ne peut pas exister :

- Si $\sigma$ ne contient aucune arête de saut, soit $\sigma$ ne contient que des arêtes positives et il ne peut pas être un cycle (car c'est un chemin montant dans $\mathcal{G}$ qui va s'arrêter dans un nœud 1, $\top$, !, & ou *ax*), soit $\sigma$ contient une arête négative et en suivant $\sigma$ à partir de cette arête, on se dirige vers le bas jusqu'à une conclusion puisqu'il n'y a pas de coupures (lemme 5.17) ce qui ne peut pas donner lieu à un cycle.
- Si $\sigma$ contient $n > 0$ arêtes de saut, on se fixe un point de départ dans $\sigma$ et on appelle $l_i$ le lien $\forall$ vers lequel est dirigée la $i^{\text{e}}$ arête de saut $a_i$ de $\sigma$. On note $X_i$ la variable propre de $l_i$. Le lien $l$ d'où part $a_{i+1}$ est situé sous $l_i$ puisque $\sigma$ n'emprunte pas d'arête de saut entre $a_i$ et $a_{i+1}$ et puisque $\mathcal{S}$ ne contient pas de coupures. La conclusion $a$ de $l$ a un type qui contient la variable libre $X_{i+1}$, par le lemme 5.17, on a une conclusion négative sous cette

arête qui ne peut pas contenir $X_{i+1}$ puisque $\mathcal{S}$ est une structure de preuve. Or, la seule manière dont $X_{i+1}$ peut disparaître entre $a$ et cette conclusion et en passant dans le nœud $l_{i+1}$, d'où on déduit que $l_{i+1}$ se trouve au-dessous de $l_i$. En remplaçant dans $\sigma$ le chemin entre $l_i$ et $l_{i+1}$ par le chemin descendant qui va de l'un à l'autre (et n'utilise pas d'arête de saut), on obtient un cycle contenant une arête de saut de moins que $\sigma$ et qui donc ne peut pas exister par hypothèse de récurrence. On en conclut que le cycle $\sigma$ ne peut pas exister.

□

5.23. Exemple
L'ajout d'arêtes de saut pour les quantificateurs $\forall$ est cependant indispensable en présence de coupures.

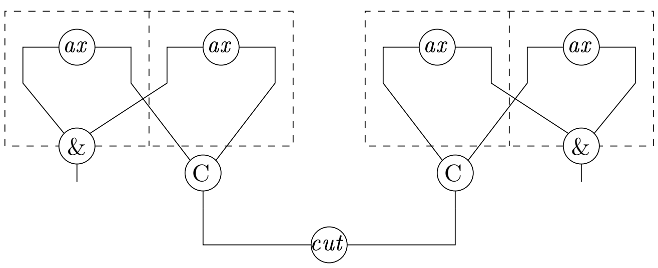



Le seul lien terminal est le lien *cut* et $(\forall XN) \parr ?N^\perp$ (qui correspond à la formule intuitionniste $N \to \forall XN$) n'est pas prouvable si $X \in N$.

### 5.1.3 Élimination des coupures

Si nous n'avons pas introduit les connecteurs additifs dans le chapitre 1, c'est qu'ils posent de nombreux problèmes pour les réseaux. L'un des problèmes concerne la séquentialité comme nous le verrons plus loin (exemples 5.41 et 5.49). Une autre difficulté probablement plus essentielle est la définition d'une procédure d'élimination des coupures confluente, de préférence pouvant réduire n'importe quelle coupure d'un réseau (on dira alors que la procédure d'élimination des coupures est *complète*).

5.24. Exemple




Il y a deux possibilités pour réduire cette coupure : soit la boîte de gauche duplique d'abord la boîte de droite, soit c'est le contraire. On obtient deux formes normales différentes.

Dans le cadre polarisé, il est possible de résoudre ce conflit puisque la symétrie est brisée : lors d'une telle coupure, l'une des prémisses est positive et l'autre est négative. On peut donc choisir un ordre particulier pour l'élimination de la coupure. Le problème qui se pose alors est de savoir quelle partie du réseau dupliquer. Dans MELL, ce problème est résolu par le fait que seules des formules commençant par ! peuvent être dupliquées et à une telle formule est toujours associée une boîte dans un réseau, c'est le candidat idéal à être dupliqué. L'introduction des connecteurs additifs d'une part et des règles structurelles généralisées d'autre part fait que l'on ne peut pas trouver de solution aussi directe. Supposons que la formule contractée soit négative, la formule à dupliquer est alors positive et on possède un candidat idéal pour la duplication : l'arbre positif. Si, par contre, la formule contractée est positive, on possède pas de tel candidat, ceci provient du fait que l'on peut calculer de manière simple le royaume d'une arête positive mais pas celui d'une arête négative.

Il en découle que l'on ne va pas être capable de définir une procédure d'élimination des coupures complète pour les réseaux de LLP. Il faut tout d'abord interdire les contractions sur des formules positives, or les seules qui puissent exister sont des contractions additives. Quand on parlera d'élimination des coupures dans les réseaux polarisés, on considérera donc des réseaux pour le sous-système taLLP de LLP qui impose que le contexte des règles & ne contienne pas de formule positive (et de même pour les règles $\top$). Le lemme suivant montre que cette contrainte n'est pas trop restrictive.

5.25. Lemme (Prouvabilité dans taLLP)
*Soit* $\vdash \Gamma$ *un séquent de* LLP sans formule positive*,* $\vdash \Gamma$ *est prouvable dans* LLP ssi $\vdash \Gamma$ *est prouvable dans* taLLP*.*

Démonstration : Puisque taLLP est un sous-système de LLP, la deuxième implication est immédiate. Passons à la première : soit $\pi$ une preuve sans coupure de $\vdash \Gamma$ dans LLP, on considère le réseau sans coupure $\mathcal{R}$ associé. Si $n$ est un nœud C positif ou $\top$ avec une conclusion positive, puisque $\mathcal{R}$ est sans conclusion positive, $n$ se trouve héréditairement au-dessus d'un nœud $?d$, on note $p_n$ le morceau de chemin allant de $n$ à ce nœud $?d$. On définit la taille de $n$ comme le nombre de nœuds non C dans $p_n$ plus la somme pour chaque nœud $\otimes$ de $p_n$ du nombre de nœuds C et $\top$ situés au-dessus de sa prémisse qui n'est pas dans $p_n$. On définit alors $\mu_{\mathcal{R}}$ comme le multi-ensemble qui à l'entier $k$ associe le nombre de nœuds C positifs ou $\top$ dans $\mathcal{R}$ de taille $k$. On montre, par récurrence sur $\mu_{\mathcal{R}}$ en utilisant l'ordre multi-ensemble, que l'on peut éliminer tous les nœuds C positifs de $\mathcal{R}$ (le cas des nœuds $\top$ à conclusion positive étant plus simple). Pour cela, on choisit un nœud C positif $n$ n'ayant aucun nœud C dans $p_n$, et on considère chaque cas de nœud situé sous $n$ (qui ne peut pas être terminal s'il est positif puisque toutes les conclusions du réseau sont négatives).

- ($?d$) On remplace ce nœud par deux nœuds $?d$ situés au-dessus des prémisses de $n$, ainsi $n$ devient un nœud C négatif. On peut appliquer l'hypothèse de récurrence car $\mu_{\mathcal{R}}$ diminue puisqu'on a supprimé un nœud C en diminuant nécessairement la taille des autres.
- ($\otimes$) Soit $\mathcal{A}$ l'arbre positif de l'autre prémisse du nœud $\otimes$, on duplique $\mathcal{A}$ que l'on place dans chacune des boîtes du nœud & associé à $n$ avec un nœud $\otimes$ entre chaque copie et la prémisse correspondante de $n$, on ajoute enfin les nœuds C négatifs nécessaires sur les feuilles des copies de $\mathcal{A}$. Soit $\mathcal{A}$ ne contient pas de nœud C et comme dans le cas précédent on peut appliquer l'hypothèse de récurrence, soit il en contient et on vérifie que $\mu_{\mathcal{R}}$ diminue également.
- ($\oplus_i, \exists$) On remplace ce nœud par deux nœuds identiques situés au-dessus des prémisses de $n$, et on applique l'hypothèse de récurrence. □

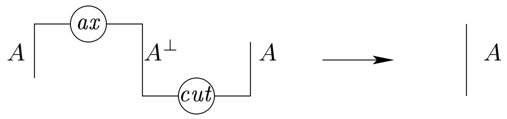
En fait, non seulement la prouvabilité est préservée mais de plus les opérations de commutation nécessaires pour transformer une preuve de LLP en preuve de taLLP ne sont pas violentes, elles préservent en tout cas la sémantique cohérente.

Pour ce qui est de l'utilisation de ces réseaux, il sera important de remarquer que, excepté pour LC, les traductions décrites dans la partie III sont à image dans taLLP.

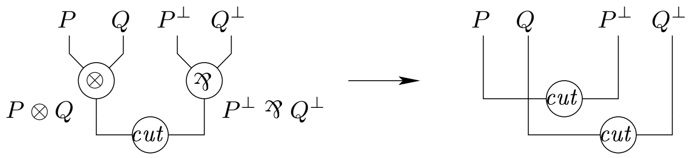
5.26. Remarque : Dans taLLP, les arbres positifs ne contiennent ni nœuds & ni nœuds $\top$.

5.27. Définition (Étapes d'élimination des coupures)
Les étapes d'élimination des coupures pour les réseaux polarisés sont les suivantes :

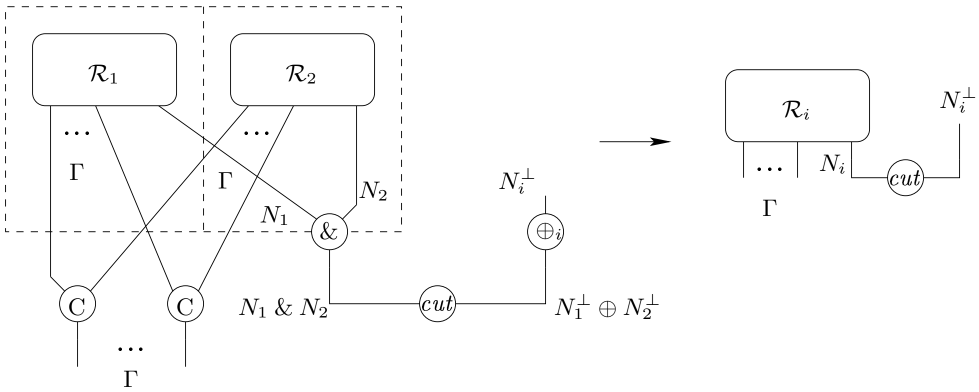
$(ax)$



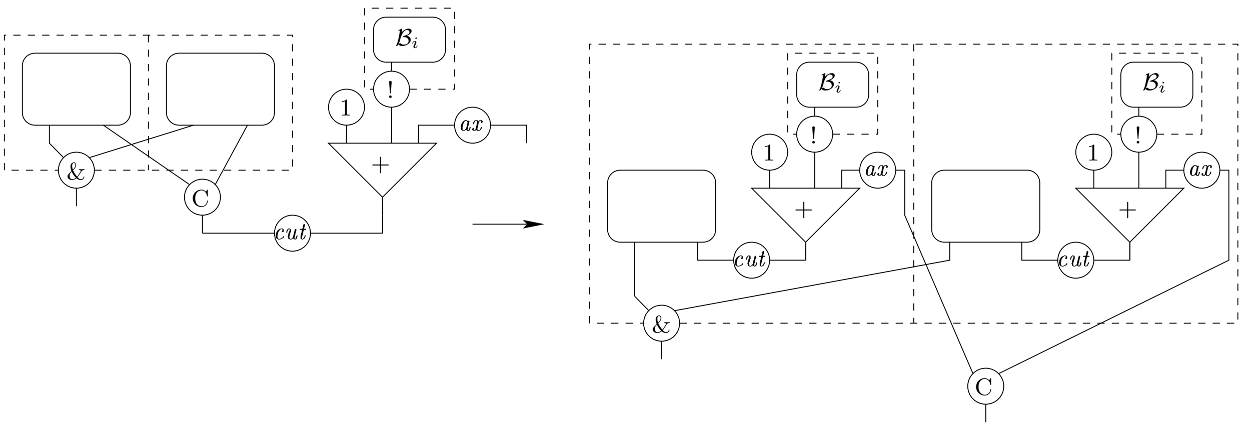

$(\otimes/\parr)$



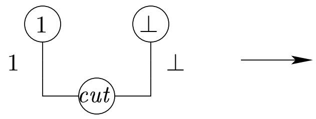

$(\&/\oplus_i)$



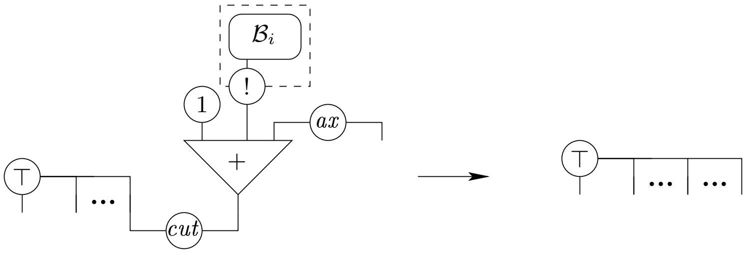

(C)




$(1/\bot)$

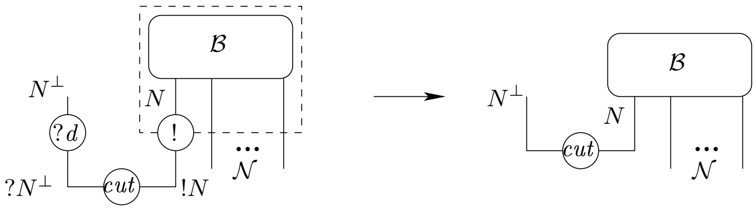


($\top$)



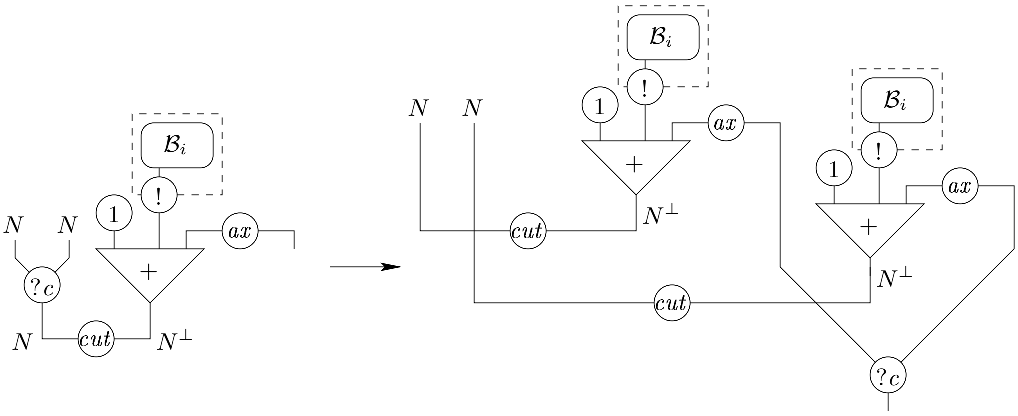
$\mathcal{B}_i$
!
1
ax
+
$\top$
...
cut
$\top$
... ...


($!/?d$)



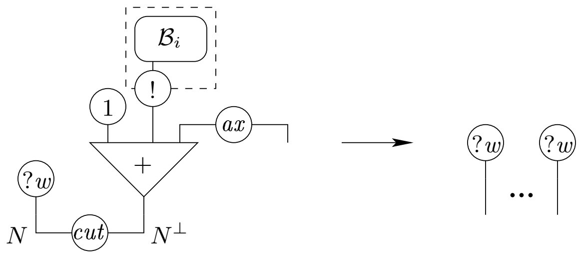
$\mathcal{B}$
$N^\perp$
N
?d
!
...
$\mathcal{N}$
$?N^\perp$
cut
$!N$
$\mathcal{B}$
$N^\perp$
N
cut
...
$\mathcal{N}$


($!/?c$)



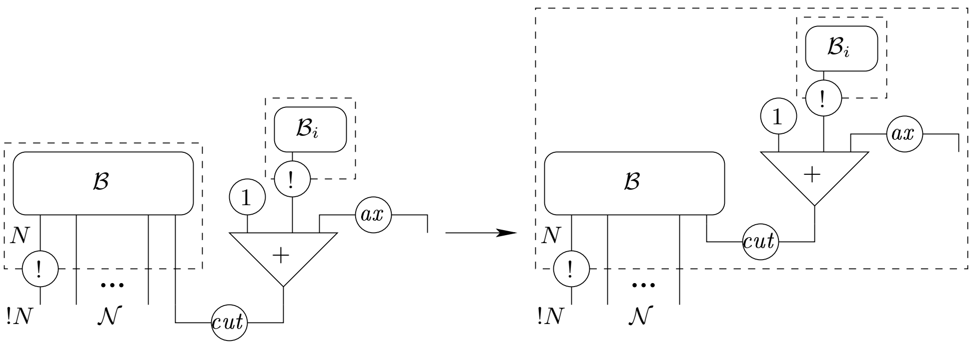
$\mathcal{B}_i$
!
1
ax
+
N
N
?c
N
cut
$N^\perp$
N
N
$\mathcal{B}_i$
!
1
ax
+
cut
$N^\perp$
$\mathcal{B}_i$
!
1
ax
+
$N^\perp$
cut
?c


($!/?w$)



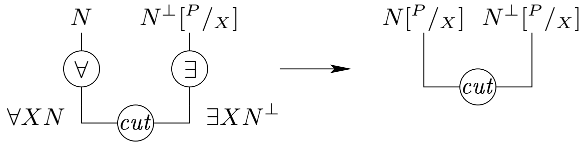
$\mathcal{B}_i$
!
1
ax
+
?w
N
cut
$N^\perp$
?w
?w
...


($!/!$)



$\mathcal{B}$
N
!
...
$!N$
$\mathcal{N}$
$\mathcal{B}_i$
!
1
ax
+
cut
$\mathcal{B}_i$
!
1
ax
+
$\mathcal{B}$
cut
N
!
...
$!N$
$\mathcal{N}$

(∀/∃)




où $P$ remplace $X$ partout dans $\mathcal{R}$.

5.28. Remarque : La définition des étapes de réduction montre bien que les contraintes sur les formules ayant le droit d'être contractées ou affaiblies et sur celles ayant le droit d'être contexte de règles de promotion doivent être cohérentes : commencer par ? dans $\mathsf{LL}$, être négative dans $\mathsf{LLP}$.

5.29. Remarque : Les opérations effectuées sur les structures de preuve par les étapes de réduction sont habituellement de deux types : soit locales (y compris les opérations de boîtes), soit globales (substitution pour les quantificateurs). On ajoute ici des opérations intermédiaires dues à l'utilisation des arbres positifs. Il faut en fait considérer ces opérations comme locales d'une part parce que les arbres positifs ne sont qu'une généralisation de la notion de boîte et d'autre part parce que la duplication, l'effacement, ... d'un arbre positif peuvent se décomposer en étapes locales.

5.30. Proposition (Préservation de la correction)
*Soit $\mathcal{R}$ un réseau de preuve polarisé et $\mathcal{R}'$ un réduit de $\mathcal{R}$, $\mathcal{R}'$ est un réseau.*

Démonstration : On considère le cas de la coupure $!/?d$ qui est le plus intéressant. Concernant les sommets initiaux, le réseau de départ a pour seul nœud initial non affaiblissement le nœud $?d$ et la boîte contient également un nœud initial non affaiblissement. Après réduction le seul nœud initial non affaiblissement du réseau est celui provenant de la boîte.

Pour l'acyclicité, s'il existe un cycle dans le réduit soit il provient d'un cycle du réseau de départ ce qui est impossible par hypothèse, soit il passe par la coupure, la boîte et une porte auxiliaire de la boîte. Dans ce cas, en remplaçant la partie contenue dans la boîte par l'arête reliant la porte auxiliaire et le nœud !, on obtiendrait un cycle dans le graphe de correction du réseau de départ. □

### 5.1.4 Propriétés

Afin de prouver de manière simple les propriétés de l'élimination des coupures pour les réseaux de $\mathsf{taLLP}$, on va passer par une traduction de $\mathsf{LLP}$ dans $\mathsf{LL}^+_{\mathrm{pol}}$ qui consiste à remettre des boîtes.

5.31. Définition (Traduction avec boîtes)
La traduction $(.)^b$ des formules positives est donnée par :

$$\begin{array}{rcl}
X^b & = & !X \\
(P \otimes Q)^b & = & !(P^b \otimes Q^b) \\
(P \oplus Q)^b & = & !(P^b \oplus Q^b) \\
1^b & = & !1 \\
0^b & = & !0 \\
(\exists XP)^b & = & !\exists XP^b \\
(!N)^b & = & !N^b
\end{array}$$

la traduction des formules négatives s'obtient par dualité.

Soit $\mathcal{R}$ un réseau polarisé, le réseau $\mathcal{R}^b$ s'obtient en ajoutant un nœud $?d$ sous chaque nœud négatif et un nœud ! sous chaque nœud positif (avec comme boîte, l'arbre positif du nœud). Chaque formule $A$ étant remplacée par sa traduction $A^b$.

5.32. REMARQUE : Soit $A^b$ la traduction de la formule $A$, les connecteurs de $A^b$ ajoutés par la traduction sont exactement les connecteurs exponentiels apparaissant sous la forme $!P$ et $?N$.

5.33. LEMME (INJECTIVITÉ DE $(.)^b$)
*La traduction $(.)^b$ de* LLP *dans* $\mathsf{LL}^+_{\text{pol}}$ *est injective.*

Démonstration : On construit une fonction $f$ de l'image de la traduction $(.)^b$ dans LLP. Pour les formules $f(A)$ est obtenue en supprimant dans $A$ les connecteurs exponentiels apparaissant sous la forme $!P$ et $?N$, on a d'après la remarque 5.32 que $f(A^b) = A$. Pour les réseaux $f$ supprime les nœuds $?d$ dont la prémisse est négative et les nœuds ! dont la prémisse est positive, on a $f(\mathcal{R}^b) = \mathcal{R}$. On déduit de l'existence de $f$ que $(.)^b$ est injective. □

Le système taLL [DJS97] est le sous-système de LL dans lequel les règles & et les règles $\top$ sont contraintes à avoir un contexte $?\Gamma$. On voit facilement que l'image de taLLP par la traduction $(.)^b$ est incluse dans taLL.

5.34. THÉORÈME (SIMULATION DE LA RÉDUCTION)
*Soit $\mathcal{R}$ un réseau polarisé, si $\mathcal{R}$ se réduit en $\mathcal{R}'$ en une étape alors $\mathcal{R}^b$ se réduit en $\mathcal{R}'^b$ en au moins une étape dans* taLL *:*

$$\begin{array}{ccc} \mathcal{R} & \longrightarrow & \mathcal{R}' \\ {\scriptstyle (.)^b}\big\downarrow & & \big\downarrow{\scriptstyle (.)^b} \\ \mathcal{R}^b & \xrightarrow{+} & \mathcal{R}'^b \end{array}$$

Démonstration : On considère chaque étape de réduction. Un nœud *cut* $n$ dans le réseau $\mathcal{R}$ a exactement un nœud correspondant $n'$ dans le réseau $\mathcal{R}^b$, la réduction de $n$ est simulée par éventuellement plusieurs réductions de $n'$ :

($ax$) Si la formule coupée est négative, l'axiome a pu être mis dans des boîtes à cause des nœuds positifs situés sous son arête positive auquel cas $n'$ à comme prémisse une arête négative qui est autant de fois porte auxiliaire. La réduction dans $\mathcal{R}^b$ consiste à d'abord effectuer les étapes !/! puis l'étape $ax$ pour obtenir $\mathcal{R}'^b$. Si la formule coupée est positive, l'étape est simulée par une étape $ax$ puisque le nœud $ax$ n'a été mis dans aucune boîte.

($\otimes/\parr$) Simulation par une étape $!/?d$ et une étape $\otimes/\parr$.

($\&/\oplus_i$) Simulation par une étape $!/?d$ et une étape $\&/\oplus_i$.

(C) Simulation par une étape $!/?c$.

($1/\bot$) Simulation par une étape $!/?d$ et une étape $1/\bot$.

($\top$) Simulation par une étape $\top$.

($\forall/\exists$) Simulation par une étape $!/?d$ et une étape $\forall/\exists$.

Les étapes exponentielles sont simulées par elles-mêmes. □

5.35. REMARQUE : Si $\mathcal{R}$ est un réseau polarisé, les boîtes de $\mathcal{R}^b$ dont le ! porte sur une formule positive (que l'on appellera *boîtes positives*) contiennent des arbres positifs. Les coupures de $\mathcal{R}^b$ sont soit des coupures exponentielles, soit des coupures $ax$, soit des coupures $\top$.

5.36. PROPOSITION (SIMULATION INVERSE)
*Soit $\mathcal{R}$ un réseau polarisé, si $\mathcal{R}^b \to \mathcal{S}$ et si $\mathcal{S}'$ est le réseau obtenu en réduisant dans $\mathcal{S}$ :*
- *les coupures $\otimes/\parr$, $\&/\oplus_i$, $1/\bot$ et $\forall/\exists$,*
- *les coupures !/! et ax situées à profondeur relative 0 dans les boîtes positives qui ne sont pas des arbres positifs.*

*Il existe un réseau polarisé $\mathcal{R}'$ tel que ${\mathcal{R}'}^b = \mathcal{S}'$ et de plus $\mathcal{R} \to \mathcal{R}'$.*

Démonstration : Une fois réduites les coupures nécessaires, on peut définir $\mathcal{R}'$ comme le réseau obtenu en supprimant dans $\mathcal{S}'$ les boîtes positives et les nœuds $?d$ situés sous des nœuds négatifs. On a alors ${\mathcal{R}'}^b = \mathcal{S}'$ et $\mathcal{R} \to \mathcal{R}'$ en réduisant la coupure réduite pour passer de $\mathcal{R}^b$ à $\mathcal{S}$. □

5.37. Corollaire (Normalisation forte)
*La réduction des réseaux polarisés est* fortement normalisante.

Démonstration : Raisonnons par l'absurde en supposant qu'il existe une suite de réductions infinie entre les réseaux polarisés $\mathcal{R}_1$, $\mathcal{R}_2$, ... Par le théorème 5.34, on en déduit que $\mathcal{R}_1^b$, $\mathcal{R}_2^b$, ... est une suite de réseaux de taLL telle que $\mathcal{R}_i^b$ se réduise en $\mathcal{R}_{i+1}^b$ en au moins une étape ce qui est impossible car il n'y a pas de suites infinies de réductions dans taLL par normalisation forte de ce système [DJS97] (qui est prouvée à l'aide d'une traduction dans MELL). □

5.38. Remarque : La normalisation forte pour LL n'a en fait jamais été entièrement démontrée (et nous ne l'utiliserons pas dans cette thèse). Les principaux éléments de la preuve se trouvent dans [Gir87, Dan90, TdF00] mais il resterait à les rassembler correctement.

5.39. Corollaire (Confluence)
*La réduction des réseaux polarisés est* confluente.

Démonstration : Soit $\mathcal{R}$ un réseau polarisé et $\mathcal{R}_1$ et $\mathcal{R}_2$ deux réduits. Par le théorème 5.34, $\mathcal{R}^b \to^+ \mathcal{R}_1^b$ et $\mathcal{R}^b \to^+ \mathcal{R}_2^b$. Par confluence de taLL [DJS97], il existe un réseau $\mathcal{S}$ tel que $\mathcal{R}_1^b \to^* \mathcal{S}$ et $\mathcal{R}_2^b \to^* \mathcal{S}$. Soit $\mathcal{S}'$ le réseau obtenu en réduisant dans $\mathcal{S}$ les coupures nécessaires comme dans la proposition 5.36, il existe un réseau polarisé $\mathcal{R}'$ tel que $\mathcal{R}_1 \to^* \mathcal{R}'$ et $\mathcal{R}_2 \to^* \mathcal{R}'$ d'où la confluence. □

5.40. Remarque : En supposant la normalisation forte, il est possible de prouver plus simplement la confluence en montrant l'unicité de la forme normale par le lemme 5.33. La preuve plus générale ci-dessus a l'intérêt de s'appliquer également dans un cadre non typé où la réduction peut ne pas terminer (voir section 11.1).

## 5.2 Réseaux à poids

Le principal défaut des réseaux polarisés *avec boîtes additives*, c'est-à-dire comme nous les avons présentés, est qu'ils conservent du calcul des séquents une certaine séquentialité entre les règles & qui n'a pas de sens calculatoire particulier (voir exemple 5.41).

Les réseaux à poids ont été introduits par Girard [Gir96] pour résoudre ce problème dans le cas de LL. Nous allons voir que l'on peut appliquer cette technique aux réseaux polarisés et qu'à nouveau on obtient un critère de correction plus simple que pour LL et une élimination des coupures plus satisfaisante. Les résultats sont principalement ceux de [Lau99] mais généralisés en différents points.

5.41. Exemple
Les deux preuves suivantes qui ne différent que par l'ordre d'application des règles & sont identifiées par la sémantique cohérente.

$$
\dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdash \Gamma, A, C\end{array} \quad \begin{array}{c}\pi_2\\ \vdash \Gamma, B, C\end{array}}{\vdash \Gamma, A \,\&\, B, C}\,\& \qquad \dfrac{\begin{array}{c}\pi_3\\ \vdash \Gamma, A, D\end{array} \quad \begin{array}{c}\pi_4\\ \vdash \Gamma, B, D\end{array}}{\vdash \Gamma, A \,\&\, B, D}\,\&}{\vdash \Gamma, A \,\&\, B, C \,\&\, D}\,\&
$$

$$
\dfrac{\dfrac{\begin{array}{c}\pi_1\\ \vdash \Gamma, A, C\end{array} \quad \begin{array}{c}\pi_3\\ \vdash \Gamma, A, D\end{array}}{\vdash \Gamma, A, C \,\&\, D}\,\& \qquad \dfrac{\begin{array}{c}\pi_2\\ \vdash \Gamma, B, C\end{array} \quad \begin{array}{c}\pi_4\\ \vdash \Gamma, B, D\end{array}}{\vdash \Gamma, B, C \,\&\, D}\,\&}{\vdash \Gamma, A \,\&\, B, C \,\&\, D}\,\&
$$

### 5.2.1 Poids et structures de preuve

5.42. Définition (Poids)
Soit $\mathcal{P}$ un ensemble de variables booléennes (appelées *poids élémentaires*), un *poids* $w$ sur $\mathcal{P}$ est un élément de l'algèbre de Boole libre engendrée par $\mathcal{P}$. On note $\leq$ l'*ordre* sur les poids, par un produit l'inf de deux poids, par une somme le sup de deux poids et par $\bar{\cdot}$ le complémentaire.

On suppose désormais fixé un ensemble de *poids élémentaires* $\mathcal{P}$ au plus dénombrable.

5.43. Définition (Valuation)
Une *valuation* $\varphi$ est une fonction de $\mathcal{P}$ dans $\{0, 1\}$.

On considérera toujours que les valuations s'étendent aux poids comme morphismes d'algèbres de Boole dans $\{0, 1\}$.

Soit $\varphi$ une valuation et $p$ un poids élémentaire, on définit $\varphi_p$ comme la valuation telle que $\varphi_p(p) \neq \varphi(p)$ et pour tout $q$ différent de $p$, $\varphi_p(q) = \varphi(q)$.

5.44. Remarque : L'ordre entre les poids peut être défini de différentes manières équivalentes :

$$
\begin{aligned}
w_1 \leq w_2 &\iff w_1 = w_1 w_2 \\
&\iff \exists w_0\, w_1 = w_0 w_2 \\
&\iff w_1 \bar{w}_2 = 0 \\
&\iff \forall \varphi\, (\varphi(w_2) = 0 \Rightarrow \varphi(w_1) = 0)
\end{aligned}
$$

5.45. Définition (Dépendance)
Soit $w$ un poids et $p$ un poids élémentaire, $w$ *dépend* de $p$ s'il existe une valuation $\varphi$ telle que $\varphi(w) \neq \varphi_p(w)$.

5.46. Définition (Structure de preuve polarisée à poids)
Une *structure de preuve polarisée à poids* est une structure de preuve au sens de la définition 1.1 construite sur les nœuds de la figure 5.1 excepté que le nœud C peut avoir plus de deux prémisses. Comme dans la définition 5.1, les types des arêtes sont des formules polarisées. On a de plus :

- Les boîtes des nœuds ! ont pour conclusion $\mathcal{N}, N$.
- À chaque nœud & est associé un poids élémentaire (distinct de ceux des autres), appelé *poids propre* du nœud. On note $\mathcal{P}$ l'ensemble de ces poids propres.
- À chaque nœud est associé un poids non nul sur $\mathcal{P}$ de manière à ce que si deux nœuds $n$ et $n'$ sont reliés par une arête alors leurs poids sont égaux excepté dans deux cas :
  - Si un nœud & a poids $w$ et poids propre $p$ alors ses prémisses ont poids $wp$ et $w\bar{p}$.
  - Si les prémisses d'un nœud C ont poids $w_1$, ..., $w_n$ alors $w_i w_j = 0$ si $i \neq j$ et le poids du nœud C est $w_1 + \cdots + w_n$.

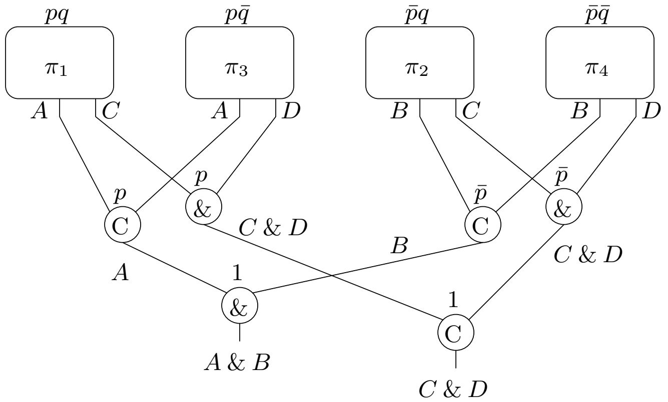
– Les poids des nœuds conclusions sont tous 1.
– Si $w$ est le poids d'un nœud & de poids propre $p$ et si $w'$ est un poids de la structure de preuve dépendant de $p$ alors $w' \leq w$.
– Si $w$ est le poids d'un nœud $\forall$ de variable propre $X$ et si $w'$ est le poids d'un nœud dépendant de $X$ alors $w' \leq w$.

5.47. Définition (Traduction du calcul des séquents)
La traduction des règles est la même que dans la définition 5.3, sauf pour la règle & :

(&) On ajoute à $\mathcal{S}_{\pi_1}$ et $\mathcal{S}_{\pi_2}$ des liens C entre les conclusions correspondant deux à deux aux formules du contexte et on ajoute un lien & entre les conclusions correspondant aux formules actives. Soit $p$ un nouveau poids élémentaire, on associe $p$ au nouveau nœud & et on multiplie les poids de $\mathcal{S}_{\pi_1}$ par $p$ et ceux de $\mathcal{S}_{\pi_2}$ par $\bar{p}$.

5.48. Remarque : Les poids ainsi obtenus sont toujours des monômes. Nous autorisons ici les poids des structures de preuve à être des éléments quelconques de l'algèbre de Boole mais, dans le cas plus général de LL, il est nécessaire de demander qu'ils soient des monômes pour obtenir le théorème de séquentialisation. Avoir des poids quelconques permet de réaliser des identification plus fines dans les réseaux et donc d'augmenter la canonicité de la représentation.

5.49. Exemple
Voici la traduction de la deuxième preuve de l'exemple 5.41 (on omet le contexte $\Gamma$) :




5.50. Remarque : Dans le cas des structures de preuve à poids, un arbre positif est formé de nœuds $\otimes$, $\oplus$, $\exists$, C positifs, $\top$, 1, ! et *ax*.

### 5.2.2 Critère de correction

5.51. Définition (Graphes de correction)
Le *graphe de correction* à profondeur 0 d'une structure de preuve à poids $\mathcal{S}$ est le graphe orienté par la *p-orientation* contenant les nœuds à profondeur 0 de $\mathcal{S}$ et dans lequel :
– On supprime les arêtes pendantes.
– Chaque lien ! dont la boîte a pour conclusions $\mathcal{N}, N$ est remplacé par un lien ! généralisé sans prémisse et avec conclusions $\mathcal{N}, !N$.
– Si $n$ est un lien & de poids propre $p$ et si $n'$ est un lien qui dépend de $p$ ou un lien C dont un lien prémisse dépend de $p$, on ajoute une arête orientée de $n'$ vers $n$, appelée *arête de saut*.
– Si $n$ est un lien $\forall$ de variable propre $X$, pour chaque lien $n'$ qui dépend de $X$, on ajoute une arête orientée de $n'$ vers $n$, appelée également *arête de saut*.

Un graphe de correction de $\mathcal{S}$ est soit son graphe de correction à profondeur 0 soit le graphe de correction à profondeur 0 d'une des boîtes de $\mathcal{S}$.

5.52. Définition (Réseau de preuve polarisé à poids)
Une structure de preuve polarisée à poids est *correcte* ou est un *réseau de preuve polarisé à poids* si :
- tous ses graphes de correction sont acycliques,
- si $w_1$, ..., $w_n$ sont les poids des nœuds initiaux non affaiblissements (*i.e.* ni $?w$ ni $\perp$) alors $w_i w_j = 0$ pour tout $i \neq j$.

On retrouve les deux lemmes de descente, la descente négative n'ayant plus de traitement spécifique pour les connecteurs additifs.

5.53. Lemme (Descente négative)
*Soit $a$ une arête négative à profondeur 0 d'une structure de preuve polarisée à poids, l'une des trois propriétés suivantes est vérifiée :*

1. *$a$ est une conclusion du réseau.*
2. *$a$ est située héréditairement au-dessus d'une coupure.*
3. *$a$ est située héréditairement au-dessus d'un nœud négatif terminal.*

5.54. Lemme (Descente positive)
*Soit $a$ une arête positive à profondeur 0 d'une structure de preuve polarisée à poids, l'une des trois propriétés suivantes est vérifiée :*

1. *$a$ est située héréditairement au-dessus d'une conclusion positive du réseau.*
2. *$a$ est située héréditairement au-dessus d'une coupure.*
3. *$a$ est située héréditairement au-dessus d'un nœud $?d$.*

5.55. Théorème (Séquentialisation)
*Une structure de preuve à poids est séquentialisable* ssi *elle est correcte.*

Démonstration : La première implication est immédiate par récurrence sur la taille de la structure de preuve.

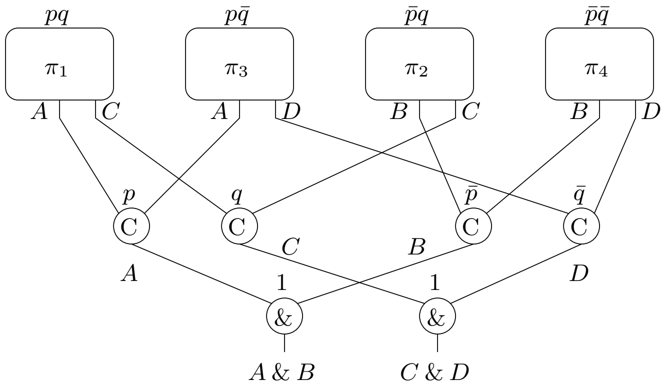
Nous démontrons la deuxième implication de manière similaire au théorème 5.19. Soit $\mathcal{R}$ un réseau de preuve polarisé à poids, on note $\mathcal{G}$ son graphe de correction à profondeur 0. La preuve se fait par récurrence sur la taille de $\mathcal{R}$ en considérant trois cas :
- Si $\mathcal{G}$ possède un nœud négatif final non $\top$, on considère chaque cas de tel nœud :
  - $(⅋, ?c, \forall)$ En supprimant ce nœud, on obtient un réseau $\mathcal{R}'$ qui est séquentialisable par hypothèse de récurrence. On en déduit que $\mathcal{R}$ est séquentialisable.
  - $(\perp, ?w)$ En supprimant ce nœud, on obtient un réseau $\mathcal{R}'$ qui est séquentialisable par hypothèse de récurrence puisque on n'a touché qu'à des nœuds initiaux affaiblissements.
  - $(\&)$ Soit $p$ le poids propre de ce nœud, on appelle $\mathcal{R}_0$ (resp. $\mathcal{R}_1$) le graphe obtenu :
    - en supprimant ce nœud dans $\mathcal{R}$,
    - en remplaçant $p$ par 0 (resp. par 1) dans chaque poids et en ne conservant que les nœuds dont le poids obtenu est non nul,
    - en remplaçant les nœuds C devenus unaires par une simple arête,
    - en conservant toutes les arêtes qui sont entre deux nœuds conservés.

On doit montrer que $\mathcal{R}_0$ et $\mathcal{R}_1$ sont des réseaux, la difficulté étant de montrer que ce sont des structures de preuve. On remarque tout d'abord que l'on a bien un graphe qui respecte le nombre de prémisses et de conclusions des nœuds ainsi que les contraintes de typage puisque si un nœud (non C) disparaît, tous les nœuds (non C) qui lui sont reliés disparaissent également d'après les contraintes sur les poids. Les nœuds C peuvent soit disparaître parce qu'ils n'ont plus qu'une ou zéro prémisse soit perdre des arêtes (dont le poids du sommet opposé s'annule) et dans ce cas le poids du nœud dans $\mathcal{R}_i$ est bien la somme des poids de ses prémisses restantes. Les poids apparaissant dans $\mathcal{R}_i$ ne dépendent que des poids propres de nœuds & présents dans $\mathcal{R}_i$. Sinon, soit $q$ un tel poids propre, $w'$ le poids dans $\mathcal{R}$ du nœud qui en dépend dans $\mathcal{R}_i$ et $w$ le poids dans $\mathcal{R}$ du nœud & correspondant, on contredit le fait que dans $\mathcal{R}$, $w' \leq w$ puisque $w$ s'annule dans $\mathcal{R}_i$ et pas $w'$. On montre de même que si un nœud $\forall$ disparaît, les occurrences de sa variable propre également. Les autres propriétés d'une structure de preuve sont facilement préservées, de même que la correction de $\mathcal{R}_0$ et $\mathcal{R}_1$.
On en déduit par hypothèse de récurrence que $\mathcal{R}_0$ et $\mathcal{R}_1$ sont séquentialisables et donc que $\mathcal{R}$ est séquentialisable.

(C) Un tel nœud n'est jamais final par définition des graphes de correction à cause des arêtes de saut. En effet, les poids des nœuds prémisses sont non nuls et différents de 1 donc dépendent d'au moins un poids propre.

– Si $\mathcal{G}$ ne possède pas de nœud négatif final non $\top$, on considère deux cas selon que $\mathcal{G}$ contient ou non des coupures. Ces cas se traitent de manière quasiment identique à la preuve du théorème 5.19. $\square$

5.56. Remarque : Pour démontrer l'intérêt des réseaux à poids il convient de donner un exemple de quotient réalisé par ces réseaux et impossible avec les réseaux à boîtes. On considère le réseau à poids suivant :




Ce réseau peut se séquentialiser en chacune des preuves de l'exemple 5.41 (on a juste omis les contextes), ce qui se voit par la parfaite symétrie entre les deux nœuds &.

Cependant la situation est complexe puisque même s'il existe un réseau identifiant les deux preuves, il existe aussi pour chacune des preuves un réseau qui ne correspond qu'à elle (exemple 5.49). Ainsi on obtient trois réseaux pour deux preuves mais avec l'avantage qu'il existe parmi eux un «bon» réseau.

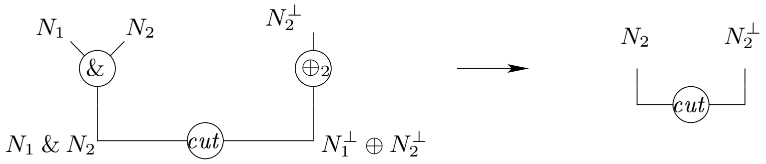
5.57. Remarque : On voit bien ici comment s'exprime la dualité positif/négatif dans le cadre des réseaux. Les nœuds négatifs sont ceux pour lesquels on définit des sauts dans le critère de correction (le cas du nœud ⅋ étant dégénéré). Les nœuds positifs préservent les royaumes : le royaume d'un nœud positif est l'union des royaumes de ses prémisses (auxquels il faut ajouter le nœud lui-même), c'est ce qui nous a permis d'introduire la notion d'arbre positif et donc l'élimination des coupures des réseaux de LLP.

### 5.2.3 Élimination des coupures

5.58. Remarque : Dans une structure de preuve à poids de taLLP, tous les nœuds d'un arbre positif ont même poids puisqu'il ne contient ni nœuds C ni nœuds &.

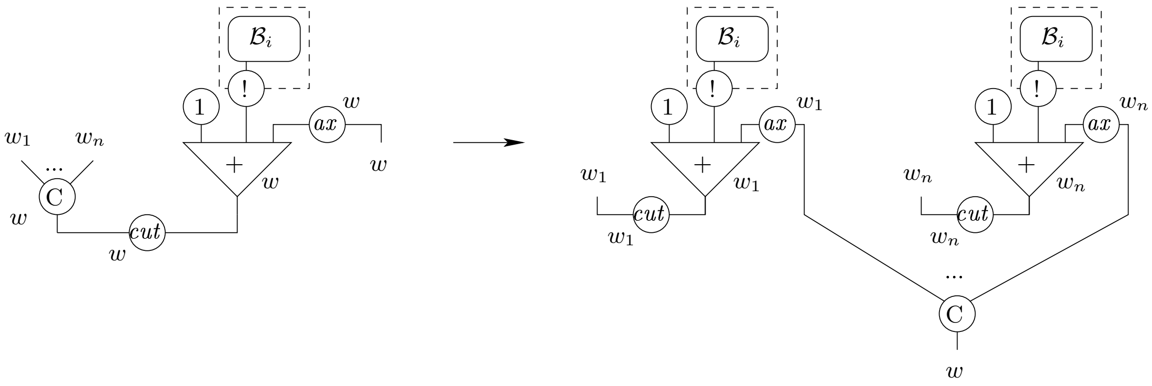
On reprend les étapes d'élimination des coupures décrites dans la définition 5.27 en corrigeant les étapes ($\&/\oplus_i$) et (C).

($\&/\oplus_i$)




si $i = 1$ (resp. $i = 2$), on remplace $p$ par 1 (resp. 0) dans les poids du réseau obtenu et on supprime les nœuds dont le poids s'annule ainsi.

(C)




On est ainsi en mesure d'éliminer n'importe quelle coupure, contrairement au cas de LL où seule une procédure paresseuse peut être définie.

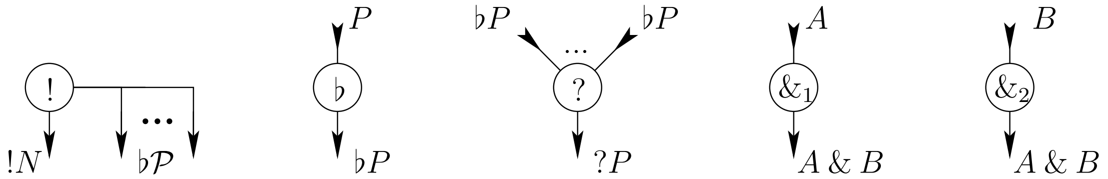
## 5.3 Réseaux par tranches

En terme de quotient réalisé sur les preuves du calcul des séquents, les réseaux additifs à boîtes et à poids ont chacun leurs défauts. Nous reprenons ici les résultats de [LTdF01] (que l'on étend aux constantes et aux quantificateurs) basés sur l'utilisation de la notion de *tranche*. Cette notion naturelle pour les réseaux additifs a été introduite par Girard dès [Gir87] mais n'a pas pu, à notre connaissance, être utilisée depuis car elle induit de réelles difficultés techniques dans le cas de LL. Nous allons montrer que pour $\mathsf{LL}_{\text{pol}}$ et en supposant dans certains cas l'absence de coupures (contrainte que l'on discutera), on peut résoudre ces difficultés et obtenir une notion de réseaux polarisés par tranches qui offre *le* quotient satisfaisant par rapport au calcul des séquents en terme de permutations de règles.

### 5.3.1 Structures de preuve par tranches

5.59. Remarque : Jusqu'ici nous avons défini les structures de preuve comme des graphes dans lesquels ont associe des sous-graphes (appelés boîtes) à certains nœuds (! et selon le cas &). Une autre possibilité, qui apparaît d'ailleurs déjà dans la définition de graphe de correction, consiste à remplacer les nœuds auxquels sont associées des boîtes dans la représentation précédente par des nœuds généralisés sans prémisse ayant autant de conclusions que la boîte associée. La séquentialité exprimée par les boîtes est ainsi encore plus explicite : une structure de preuve est constituée du graphe à profondeur 0 contenant des nœuds généralisés auxquels sont récursivement associées des structures de preuve. Cette présentation est un peu plus souple et plus pratique pour les questions de séquentialisation mais ne correspond pas tout à fait à la représentation naturelle des réseaux et complique les définitions d'élimination des coupures qui reste tout de même le point essentiel. C'est pourquoi nous avons choisi jusqu'ici une présentation avec boîtes comme sous-réseaux. Pour les réseaux par tranches cette présentation devenant beaucoup plus compliquée à gérer, nous utiliserons la présentation profondeur par profondeur.

5.60. Définition (Tranche)
Une *tranche* est une structure de preuve au sens de la définition 1.1 construite sur les nœuds de la figure 5.1 excepté que les nœuds !, $?c$, $?w$, $?d$, &, C et $\top$ n'apparaissent pas mais par contre on utilise les nouveaux nœuds (avec la *s-orientation* ici) :



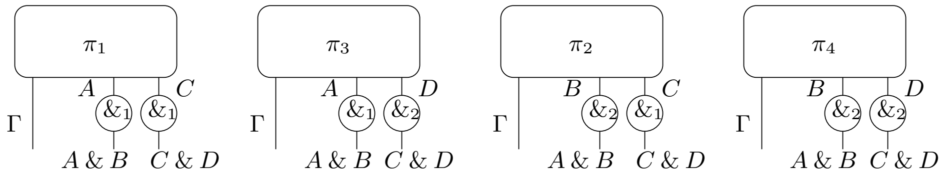

Le type d'une arête peut être soit une formule polarisée soit une formule positive précédée du symbole $\flat$. On demande de plus que les axiomes n'introduisent que des atomes (structure *$\eta$-expansée*).

Les nœuds ? à un nombre arbitraire (y compris 0) de prémisses (voir [Reg92]) permettent de remplacer les nœuds $?w$ et $?c$ en obtenant directement le quotient de l'*équivalence structurelle* (définition 1.6). Les formules $\flat P$ correspondent aux $[P]$ de [Gir96], et à la construction duale du $\sharp$ du chapitre 3.

5.61. Définition (Structure de preuve par tranches)
Une *structure de preuve par tranches* est soit un ensemble de tranches sans nœud ! soit un ensemble de tranches ayant, récursivement, une structure de preuve par tranches (de conclusions $N, \flat\mathcal{P}$) associée à chaque nœud ! de conclusions $!N, \flat\mathcal{P}$. Si $\mathcal{S}$ est l'ensemble de tranches $\{s_1, \ldots, s_n\}$, de conclusions respectives $\Gamma_i, \flat\Delta_i$, on demande que les $\Gamma_i$ soient égaux (notés $\Gamma$) et les conclusions de $\mathcal{S}$ sont $\Gamma, \flat\Delta_1, \ldots, \flat\Delta_n$.

5.62. Définition (Traduction des structures de preuve à poids)
Soit $\mathcal{S}$ une *structure de preuve à poids*, on définit l'ensemble de tranches $\mathcal{T}$ contenant, pour chaque valuation $\varphi$ des poids à profondeur 0, la tranche $\varphi(\mathcal{S})$ (et récursivement à chaque profondeur). Pour construire $\varphi(\mathcal{S})$, on remplace les nœuds $?d$ par des nœuds $\flat$ et les *arbres exponentiels* sont remplacés par des nœuds ?. Si $\varphi(\mathcal{S})$ contient un nœud $\top$ à profondeur 0, elle n'apparaît pas dans $\mathcal{T}$. Enfin on remplace les liens *ax* par leur *$\eta$-expansion*.

5.63. Exemple
Nous reprenons les exemples 5.41 et 5.49 traduits en réseau par tranches :




5.64. Remarque : Les structures de preuve par tranches qui nous intéressent doivent pouvoir correspondre à des preuves de logique, elles ne possèdent donc pas de conclusion $\flat$. Nous nous restreindrons désormais à de telles structures de preuve par tranches.

### 5.3.2 Correction

Pour les résultats de correction, on va se restreindre aux structures de preuve sans coupure. En effet, en présence de coupures, la notion de recollement de tranches devient impossible à définir. Cette contrainte n'empêche cependant pas de calculer, il suffit de partir de structures sans coupure, d'introduire des coupures et de normaliser jusqu'au bout pour obtenir à nouveau une structure sans coupure (ceci correspond à un système de réduction à grands pas alors qu'on utilise habituellement en logique des réductions à petits pas), voir remarque 5.80.

5.65. Remarque : C'est également pour des questions de recollement que l'on est amené à ne considérer que des structures $\eta$-expansés. Sinon on pourrait avoir un axiome non atomique dans une tranche et une $\eta$-expansion dans une autre ce qui rendrait le recollement compliqué.

5.66. Définition (Nœuds équivalents)
Soient $\mathcal{S}$ une structure de preuve par tranches sans coupure et $s_1$ et $s_2$ deux tranches de $\mathcal{S}$. Soit $n_1$ (resp. $n_2$) un nœud logique négatif de $s_1$ (resp. $s_2$), on définit la relation d'équivalence $n_1 \equiv n_2$ par récurrence sur le nombre de nœuds au-dessous de $n_1$ :
- Si $n_1$ est conclusion de $s_1$, si $n_2$ est conclusion de $s_2$ et si leurs conclusions sont les mêmes alors $n_1 \equiv n_2$.
- Si $n_1$ est prémisse d'un nœud $m_1$ unaire, $n_2$ est prémisse d'un nœud $m_2$ unaire, $m_1 \equiv m_2$ et $m_1$ et $m_2$ sont deux nœuds du même type alors $n_1 \equiv n_2$.
- Si $n_1$ est prémisse gauche (resp. droite) d'un nœud $m_1$ binaire, $n_2$ est prémisse gauche (resp. droite) d'un nœud $m_2$ binaire, $m_1 \equiv m_2$ et $m_1$ et $m_2$ sont deux nœuds du même type alors $n_1 \equiv n_2$.

5.67. Remarque : Un nœud $\&_1$ et un nœud $\&_2$ peuvent très bien être équivalents (c'est le seul cas où des nœuds équivalents ne sont pas du même type) mais leurs prémisses ne le sont pas.

La définition de l'équivalence de nœuds négatifs est rendue possible par la forme très particulière des structures de preuve pour $\mathsf{LL}_{\mathrm{pol}}$ : à profondeur 0, sous un nœud logique négatif ne peuvent se trouver que des nœuds logiques négatifs. L'idée intuitive est alors tout simplement que deux nœuds sont équivalents s'ils correspondent à la même occurrence de connecteur (sous la portée d'aucun connecteur exponentiel) dans une des formules conclusion de l'ensemble de tranches.

5.68. Définition (Poids dans une tranche)
Soit $\mathcal{S}$ une structure de preuve par tranches sans coupure de conclusions $\Gamma$, on note $\&^1$, ..., $\&^n$ les classes d'équivalence de nœuds $\&_i$ à profondeur 0 de $\mathcal{S}$ (ou, de manière équivalente, les occurrences de connecteurs & dans $\Gamma$ sous la portée d'aucun connecteur exponentiel). À chaque $\&^i$, on associe un poids élémentaire $p_i$. Le poids de la tranche $s$ de $S$ est :

$$w(s) = \prod_{\&_1^i \in s} p_i \prod_{\&_2^i \in s} \bar{p}_i$$

Le poids de $S$ est :

$$w(S) = \sum_{s \in S} w(s)$$

Le poids d'un nœud $n$ (modulo équivalence) est :

$$w(n) = \sum_{n \in s} w(s)$$

On définit le poids de $\top$ dans une formule $A$ de $\mathcal{S}$ par récurrence sur la taille de cette formule :
- si $A$ est positive, $w_\top(A) = 0$
- $w_\top(X^\perp) = 0$
- $w_\top(\perp) = 0$
- $w_\top(\top) = 1$
- $w_\top(N \parr M) = w_\top(N) + w_\top(M)$
- $w_\top(N \,\&\, M) = p w_\top(N) + \bar{p} w_\top(M)$ où $p$ est le poids propre du nœud & correspondant
- $w_\top(\forall X N) = w_\top(N)$
- $w_\top(?P) = 0$

Le poids de $\top$ dans $\mathcal{S}$ est :

$$w_\top(S) = w_\top(\parr \Gamma)$$

5.69. Définition (Réseau de preuve polarisé par tranches)
Une structure de preuve polarisée par tranches $\mathcal{S}$ est *acceptable* si tous ses graphes de correction possèdent un unique nœud initial non affaiblissement (*i.e.* ni $?w$ ni $\perp$) et si $w(S) + w_\top(S) = 1$ et pour toutes tranches $s \neq t$ de $\mathcal{S}$, $w(s)w(t) = 0$.

Une structure de preuve polarisée par tranches $\mathcal{S}$ est *correcte* ou est un *réseau de preuve polarisé par tranches* si elle est acceptable et si tous ses graphes de correction sont acycliques.

5.70. Théorème (Séquentialisation)
*Une structure de preuve par tranches sans coupure est séquentialisable* ssi *elle est acceptable.*

Démonstration : La première implication ne pose pas de problème. La deuxième se démontre de manière comparable aux théorèmes 5.19 et 5.55. On raisonne par récurrence sur la taille de la structure de preuve par tranches $\mathcal{S}$, c'est-à-dire la somme des nombres de nœuds de chaque tranche.
- Si une tranche de $\mathcal{S}$ a un nœud $\parr$ terminal alors les autres tranches également et on peut retirer ce nœud dans toutes les tranches.
- De même pour un nœud $\forall$ ou $\perp$.
- Si une tranche de $\mathcal{S}$ a un nœud & terminal de conclusion $N \,\&\, M$ alors les autres tranches ont un nœud & équivalent. Soit $\mathcal{S}_1$ (resp. $\mathcal{S}_2$) l'ensemble des tranches de $\mathcal{S}$ contenant ce nœud sous la forme $\&_1$ (resp. $\&_2$) dans lesquelles on supprime ce nœud. Si $p$ est le poids propre du &, on a $w(\mathcal{S}) = pw(\mathcal{S}_1) + \bar{p}w(\mathcal{S}_2)$ et $w_\top(\mathcal{S}_1) + \bar{p}w_\top(M) = w_\top(\mathcal{S}) + \bar{p}w_\top(N)$ donc $p = pw(\mathcal{S}) + pw_\top(S) = pw(\mathcal{S}_1) + pw_\top(S) = pw(\mathcal{S}_1) + pw_\top(\mathcal{S}_1)$ d'où on déduit que $w(\mathcal{S}_1) + w_\top(\mathcal{S}_1) = 1$ puisqu'ils ne contiennent pas $p$ (et de même pour $\mathcal{S}_2$). On peut donc appliquer l'hypothèse de récurrence à $\mathcal{S}_1$ et $\mathcal{S}_2$, et on en déduit que $\mathcal{S}$ est séquentialisable.

Si aucune tranche de $\mathcal{S}$ n'a de nœud négatif logique terminal, alors $\mathcal{S}$ ne contient aucun nœud & sinon au-dessous on trouverait un nœud négatif logique terminal. On en déduit que toute tranche a poids 1 et donc que $\mathcal{S}$ contient au plus une tranche puisqu'elle est acceptable. Si $\mathcal{S}$ est réduite à une tranche, la séquentialisation ne pose alors aucun problème en supprimant d'abord les nœuds $\flat$ et $?$.

Si $\mathcal{S}$ ne contient aucune tranche alors, par hypothèse, $w_\top(\mathcal{S}) = 1$. On raisonne par récurrence sur la taille de la conclusion $\Gamma$ de $\mathcal{S}$.
- Si $\Gamma$ ne contient pas de formules négatives, $w_\top(\mathcal{S}) = 0$.
- Si $\Gamma$ ne contient que des formules négatives $X^\perp$, $?P$ ou $\perp$, alors $w_\top(\mathcal{S}) = 0$.
- Si $\Gamma$ contient une formule $\top$, alors $\mathcal{S}$ se séquentialise en une règle $\top$ introduisant ses conclusions.
- Si $\Gamma$ ne contient pas de formule $\top$ mais une formule $N \parr M$ (resp. $N \& M$ ou $\forall XN$), on applique l'hypothèse de récurrence en enlevant le $\parr$ (resp. $\&$ ou $\forall$) et on ajoute une règle $\parr$ (resp. $\&$ ou $\forall$) à la (ou les) preuve(s) obtenue(s). □

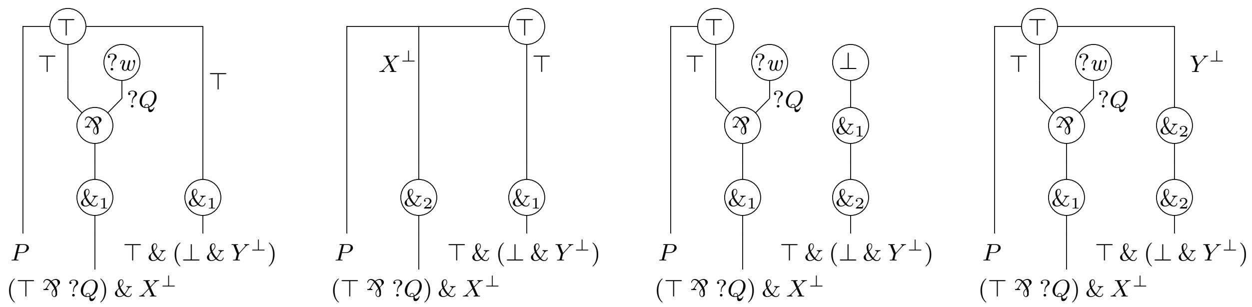
Nous pouvons également établir un lien direct entre les structures de preuve par tranches et les structures de preuve à poids.

5.71. Définition ($\top$-complément)
Soit $\mathcal{S}$ une structure de preuve par tranches de conclusions $\mathcal{N}, P$, un $\top$-*complément* de $\mathcal{S}$ est une structure de preuve par tranches définie de la manière suivante :

Pour chaque forêt (formée de nœuds $\parr$, $\&_j$ et $\forall$) des formules de $\mathcal{N}$ dont au moins une feuille est de type $\top$, on construit une tranche de conclusions $\mathcal{N}, P$ en ajoutant un nœud $\perp$ au-dessus de chaque feuille de type $\perp$, un nœud $?w$ au-dessus de chaque feuille de type $?Q$, et un nœud $\top$ de conclusion principale l'une des feuilles de type $\top$ et de conclusions auxiliaires les autres $\top$, les feuilles $X^\perp$ et $P$.

5.72. Remarque : La construction du $\top$-complément n'est pas canonique puisque la porte principale des nœuds $\top$ est arbitraire si plusieurs choix sont possibles.

5.73. Exemple
Un $\top$-complément pour une structure de preuve par tranches $\mathcal{S}$ de conclusions $P, (\top \parr ?Q) \& X^\perp, \top \& (\perp \& Y^\perp)$ (où $p$ est le poids propre du premier $\&$, $q$ celui du deuxième et $r$ celui du troisième) est :




5.74. Lemme (Poids d'un $\top$-complément)
*Soit $\mathcal{S}$ une structure de preuve par tranches, le poids d'un $\top$-complément de $\mathcal{S}$ est $w_\top(\mathcal{S})$.*

Démonstration : Soit $\mathcal{N}, P$ les conclusions de $\mathcal{S}$, on raisonne par récurrence sur la taille de $\mathcal{N}$ :
- Si $\mathcal{N}$ est vide, le $\top$-complément aussi puisqu'il n'y a pas de $\top$.
- Si $\mathcal{N} = \mathcal{N}', N \parr M$, le poids d'un $\top$-complément de $\mathcal{S}$ et le même que celui d'un $\top$-complément de la structure correspondante de conclusions $\mathcal{N}', N, M$.
- Si $\mathcal{N} = \mathcal{N}', \forall XN$ ou $\mathcal{N} = \mathcal{N}', \perp$ ou $\mathcal{N} = \mathcal{N}', ?Q$, de même.
- Si $\mathcal{N} = \mathcal{N}', N \& M$, soient $\mathcal{S}_1$ et $\mathcal{S}_2$ les deux structures correspondant à $\mathcal{S}$ de conclusions $\mathcal{N} = \mathcal{N}', N$ et $\mathcal{N} = \mathcal{N}', M$, tout $\top$-complément de $\mathcal{S}$ est obtenu à partir d'un $\top$-complément de $\mathcal{S}_1$ de poids $w_1$ et d'un $\top$-complément de $\mathcal{S}_2$ de poids $w_2$ en ajoutant des nœuds $\&_1$ sous

la conclusion $N$ des tranches du premier et des $\&_2$ sous la conclusion $M$ des tranches du second. On en déduit que le poids du $\top$-complément est $pw_1+\bar{p}w_2$ où $p$ est la variable propre associée au & de $N\&M$, or par hypothèse de récurrence, $w_1 = w_\top(\mathcal{S}_1)$ et $w_2 = w_\top(\mathcal{S}_2)$ ce qui permet de conclure que le poids d'un $\top$-complément de $\mathcal{S}$ est $pw_\top(\mathcal{S}_1)+\bar{p}w_\top(\mathcal{S}_2) = w_\top(\mathcal{S})$.
- Si $\mathcal{N} = \mathcal{N}', \top$, le poids d'un $\top$-complément est 1 puisque toutes les forêts de $\mathcal{N}$ contiennent une feuille $\top$.
- Si $\mathcal{N} = X_1^\perp, \ldots, X_n^\perp$, aucune forêt de $\mathcal{N}$ ne contient de feuille $\top$ donc le $\top$-complément est vide (*i.e.* de poids 0). □

5.75. Proposition (Recollement)
*Une structure de preuve par tranches sans coupure acceptable correspond à une structure de preuve à poids.*

Démonstration : On construit à partir de la structure de preuve par tranches acceptable $\mathcal{S}$, une structure de preuve à poids en recollant les nœuds équivalents.

On note $\bar{\mathcal{S}}$ l'ensemble des tranches de $\mathcal{S}$ et d'un $\top$-complément de $\mathcal{S}$ de conclusions $\Gamma, \flat\Delta_i$, $1 \leq i \leq k$. On considère le graphe $\mathcal{G}$ dont les nœuds sont :
- les nœuds ni $⅋$, ni &, ni $\forall$, ni $\perp$ des tranches de $\bar{\mathcal{S}}$,
- les classes d'équivalence des nœuds logiques négatifs non $\top$ (*i.e.* $⅋$, &, $\forall$ et $\perp$),
- les nœuds C nécessaires pour recoller les conclusions des nœuds $\top$ du $\top$-complément de $\mathcal{S}$ qui se correspondent,
- les nœuds C nécessaires pour recoller les conclusions de type $P$ et $?P$ provenant de différentes tranches et correspondant à la même conclusion de l'ensemble de tranches,

De plus, les nœuds $\flat$ sont remplacés par des nœuds $?d$ et les nœuds ? sont décomposés en nœuds $?c$ et $?w$. Enfin soit $i$ une des tranches de $\bar{\mathcal{S}}$, pour chaque $1 \leq j \leq k$ tel que $j \neq i$ on ajoute des nœuds $?w$ sur les $?\Delta_j$ et des nœuds C à $k$ prémisses entre ces $k-1$ occurrences de chaque formule de $?\Delta_j$ et l'occurrence provenant de la tranche $i$.

Les poids propres des nœuds & et les poids des nœuds sont obtenus comme dans la définition 5.68.

On montre que $\mathcal{G}$ est une structure de preuve à poids :
- Si $n$ est un nœud non & de $\mathcal{G}$, il a nécessairement le bon nombre de prémisses et il vérifie les contraintes de typage.
- Si $n$ est un nœud &, il existe une tranche dans laquelle il apparaît sous la forme $\&_1$ et une dans laquelle il apparaît sous la forme $\&_2$. Sinon on suppose qu'il n'apparaît que sous la forme $\&_1$, on a $\sum_{s\in\bar{\mathcal{S}}} w(s)$ qui se décompose en une somme pour les tranches où le lien apparaît et celles où il n'apparaît pas :

$$\sum_{s\in\bar{\mathcal{S}}} w(s) = \sum_{\substack{s\in\bar{\mathcal{S}} \\ \&_1\in s}} w(s) + \sum_{\substack{s\in\bar{\mathcal{S}} \\ \&\notin s}} w(s)$$

  Les termes des sommes sont tous des monômes (dont les produits deux à deux sont nuls) et ceux de la première somme contiennent $p$ et ceux de la seconde ne contiennent ni $p$ ni $\bar{p}$. Ceci est impossible puisque $\sum_{s\in\bar{\mathcal{S}}} w(s) = w(\mathcal{S}) + w_\top(\mathcal{S}) = 1$ (le poids du $\top$-complément étant $w_\top(\mathcal{S})$ par le lemme précédent).
- Les poids de deux nœuds reliés par une arête sont les mêmes puisqu'ils proviennent de la même tranche sauf pour :
  - les nœuds & dont les deux prémisses proviennent de tranches différentes et le poids du nœud est, par définition, la somme des poids des prémisses ;

– les nœuds C ajoutés entre les nœuds $\top$ qui ont également comme poids la somme des poids des prémisses ;
– les nœuds C ajoutés sous les conclusions $?P$ qui ont poids 1 qui est la somme des poids de leurs prémisses.

– Les poids des conclusions sont bien 1.
– Si le poids $w'$ d'un nœud $n'$ dépend d'un poids propre $p$ d'un nœud $n$ (de type &) de poids $w$, on doit montrer que $w' \leq w$, c'est-à-dire que si $n'$ apparaît dans une tranche alors $n$ également :
  – Si $n'$ est un nœud logique négatif (non $\top$), il n'y a que des nœuds logiques négatifs au-dessous de lui. On raisonne par récurrence sur le nombre de nœuds & situés au-dessous. S'il n'y en a aucun son poids est 1 et ne dépend pas de $p$. S'il y en a $k+1$, on considère le premier & situé sous $n'$, si c'est $n$ alors $w' \leq w$ puisque le poids augmente en descendant et si c'est $n'' \neq n$ de poids $w''$ et de poids propre $q$ alors $w' = qw''$ ou $w' = \bar{q}w''$ donc $w''$ dépend de $p$ et par hypothèse de récurrence $w'' \leq w$ d'où $w' \leq w$.
  – Si $n'$ est un nœud positif ou un nœud $\top$, il n'apparaît que dans une tranche (on n'a fait aucun recollement entre les nœuds de ce type) donc son poids est un monôme et il dépend de $p$ si et seulement si $n$ apparaît dans cette tranche donc $w' \leq w$.
– Si un nœud $n'$ de poids $w'$ dépend de la variable propre $X$ du nœud $n$ (de type $\forall$) de poids $w$, d'après la définition 5.1, si $n'$ apparaît dans une tranche alors $n$ apparaît également dans cette tranche donc $w' \leq w$. □

5.76. Remarque : On a été obligé de se restreindre à $\mathsf{LL}_{\text{pol}}$ pour pouvoir définir le critère de correction et le recollement qui posent problème en présence des règles structurelles généralisées. En utilisant une notion de réseaux avec formules localisées [Gir01], on devrait pouvoir étendre ces notions à $\mathsf{LLP}$.

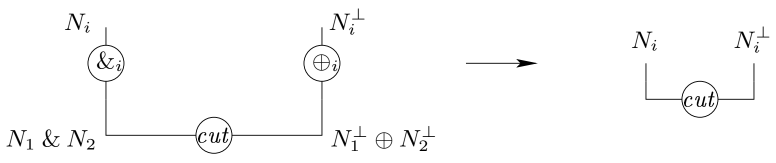
5.77. Remarque : Puisque les structures de preuve par tranches sans coupure acceptables sont séquentialisables, elles correspondent à des structures de preuve à poids acycliques et elles sont donc correctes. Ceci nous donne un analogue du corollaire 5.22 pour les réseaux à poids et les réseaux par tranches.

### 5.3.3 Élimination des coupures

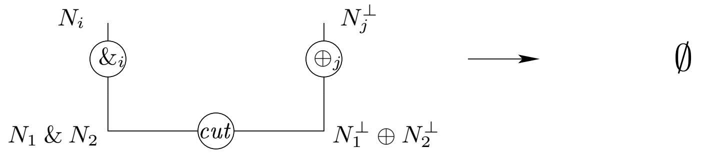
Pour des raisons de recollement, il ne nous a pas été possible de considérer les réseaux par tranches pour $\mathsf{LLP}$ mais seulement pour $\mathsf{LL}_{\text{pol}}$. En ce qui concerne l'élimination des coupures, il n'est pas nécessaire d'ajouter de contraintes concernant les formules positives dans le contexte des règles additives, c'est un des intérêts des tranches.

Les nouvelles étapes intéressantes concernent les coupures additives :

$(\&_i/\oplus_i)$



$(\&_i/\oplus_j)\ \ i \neq j$




La tranche est supprimée de l'ensemble de tranches.

Les étapes d'élimination $\top$ ont disparu puisque les nœuds $\top$ n'apparaissent pas dans les structures de preuve par tranches.

5.78. Proposition (Confluence et normalisation forte)
*L'élimination des coupures dans les réseaux polarisés par tranches est* confluente *et* fortement normalisante.

Démonstration : Il s'agit ici d'une conséquence directe de la confluence et de la normalisation forte pour MELL.

La confluence étant vraie dans un cadre non typé, il suffit de remarquer que l'on peut remplacer les nœuds & par des nœuds $\forall$ et les nœuds $\oplus$ par des nœuds $\exists$ sans changer la géométrie.

La normalisation forte s'obtient facilement à partir de celle de MELL puisque l'étape $(\&_i/\oplus_i)$ fait diminuer strictement la taille du réseau et l'étape $(\&_i/\oplus_j)$ stoppe la réduction. □

Question ouverte : Soient $\mathcal{R}_1$ et $\mathcal{R}_2$ deux réseaux polarisés par tranches sans coupure ayant la même sémantique cohérente, $\mathcal{R}_1$ et $\mathcal{R}_2$ sont-ils égaux ?

Une réponse positive est donnée dans [LTdF01] avec une contrainte sur les formules correspondant aux types du $\lambda$-calcul avec paires. Le résultat plus général énoncé ici paraît tout à fait vraisemblable.

Cette question d'*injectivité de la sémantique cohérente* par rapport aux réseaux a été étudiée en l'absence de connecteurs additifs par L. Tortora de Falco [TdF01].

5.79. Exemple
Pour résumer les exemples 5.41, 5.49 et 5.63, on a le diagramme de séquentialisation suivant :

$$\begin{tikzcd}
\text{calcul des séquents} & \text{réseaux à boîtes} & \text{réseaux à poids} & \text{réseaux par tranches} \\
\pi_1 & \mathcal{R}_1^b \arrow[l] & \mathcal{R}_1^p \arrow[l] & \\
 & & \mathcal{R}_3^p \arrow[ul] \arrow[dl] & \mathcal{R}^t \arrow[l] \arrow[ul] \arrow[dl] \\
\pi_2 & \mathcal{R}_2^b \arrow[l] & \mathcal{R}_2^p \arrow[l] &
\end{tikzcd}$$

5.80. Remarque : La nécessité de pouvoir recoller des ensembles de tranches ne nous permet pas de considérer véritablement des tranches avec coupures. La définition d'un critère de correction pour des ensembles de tranches avec coupures paraît difficile à donner.

Pour pouvoir calculer avec des ensembles de tranches, on est obligé de considérer la méthode suivante : un programme est un objet sans coupure (sinon il suffit de calculer sa forme normale), c'est au moment de lui passer ses arguments que l'on est obligé d'introduire des coupures mais ces coupures sont *externes*, ce n'est qu'au cours de la réduction qu'on les retrouvera au milieu des objets. Si on pousse la réduction jusqu'à son terme, on aboutit à nouveau à un objet sans coupure que l'on sait étudier.

Concrètement, si on cherche à calculer la valeur de $f(n)$ où $f$ est de type $\mathbb{N} \to \mathbb{N}$ et $n$ de type $\mathbb{N}$, on introduit une coupure entre $f$ et $n$ et on l'élimine jusqu'à obtenir un terme de type $\mathbb{N}$ sans coupure. Or pour un terme sans coupure, on est capable d'analyser la valeur.

Cette approche est certes moins souple que celle généralement utilisée en logique et qui consiste à pouvoir introduire les coupures à l'intérieur des termes. Cette distinction est tout à fait comparable à celle entre sémantique opérationnelle à grand pas et à petits pas que l'on rencontre en informatique. La sémantique à grand pas définit une relation entre programmes (qui sont ici les termes sans coupure coupés entre eux) et valeurs (les termes sans coupure) alors que la sémantique à petits pas définit une relation de réduction de programme en programme (étapes élémentaires de réduction donnant lieu à des coupures à l'intérieur des termes).

En logique, il existe d'autres contextes où l'on a été obligé d'imposer une telle restriction, par exemple :

- composition des transformations dinaturelles [GSS92] ;
- réduction des dessins en ludique [Gir01].

# Chapitre 6

# Sémantiques statiques de LLP

Historiquement, la première sémantique de LL est la sémantique cohérente. Adaptée à LC [Gir91a] par les *espaces de corrélation*, elle fournit naturellement une sémantique cohérente pour LLP. Une autre direction pour la sémantique dénotationnelle des systèmes logiques est l'approche catégorique. Nous montrons que les *catégories de contrôle* [Sel01] introduites par P. Selinger fournissent (de deux manières différentes par dualité) une notion de modèle catégorique de LLP. Dans ces deux cas les définitions des objets sémantiques étaient connues avant ce travail et sont reprises telles que données par leurs auteurs. Les nouveautés de ce chapitre consistent à montrer que cela donne lieu à des modèles dénotationnels de LLP.

Ces sémantiques sont traditionnellement appelées *statiques* par opposition à la géométrie de l'interaction et à la sémantique des jeux que l'on considère comme *dynamiques* du fait de la définition de la composition et de la temporalité explicite (en ce qui concerne les jeux en tout cas).

## 6.1 Espaces de corrélation

Nous reprenons les définitions données dans [Gir91a] pour les espaces de corrélations.

6.1. Définition (Espace de corrélation)
Un *espace de corrélation négatif* est un triplet $(\mathcal{N}, \eta, \mu)$ où $\mathcal{N}$ est un *espace cohérent*, $\eta$ est une clique de $\mathcal{N}$ et $\mu$ est une clique de $(\mathcal{N} \parr \mathcal{N}) \multimap \mathcal{N}$. Les cliques $\eta$ et $\mu$ doivent vérifier les équations suivantes :

(neutralité) $\forall x \in |\mathcal{N}|, \exists y \in \eta, ((x, y), x) \in \mu$

(commutativité) $\forall ((x, y), z) \in \mu, ((y, x), z) \in \mu$

(associativité) $\forall ((x, y), u) \in \mu, \forall ((u, z), t) \in \mu, \exists v \in |\mathcal{N}|, ((y, z), v) \in \mu \wedge ((x, v), t) \in \mu$

Un *espace de corrélation positif* est un triplet $(\mathcal{P}, \varepsilon, \delta)$ où $\mathcal{P}$ est un *espace cohérent*, $\varepsilon$ est une anti-clique de $\mathcal{P}$ et $\delta$ est une clique de $\mathcal{P} \multimap (\mathcal{P} \otimes \mathcal{P})$ tels que :

(neutralité) $\forall x \in |\mathcal{P}|, \exists y \in \varepsilon, (x, (x, y)) \in \delta$

(commutativité) $\forall (x, (y, z)) \in \delta, (x, (z, y)) \in \delta$

(associativité) $\forall (u, (x, y)) \in \delta, \forall (t, (u, z)) \in \delta, \exists v \in |\mathcal{P}|, (v, (y, z)) \in \delta \wedge (t, (x, v)) \in \delta$

6.2. Remarque : La propriété d'associativité permet d'écrire dans un espace de corrélation négatif : $a_1 + \cdots + a_n \mapsto a$ s'il existe $u_1$, ..., $u_{n-2}$ tels que $((a_1, u_1), a) \in \mu$, pour $1 \leq i \leq n-2$ on ait $((a_i, u_i), u_{i-1}) \in \mu$ et $((a_{n-1}, a_n), u_{n-2}) \in \mu$. De la même manière, dans un espace de corrélation positif, on note : $a \hookleftarrow a_1 + \cdots + a_n$ s'il existe $u_1$, ..., $u_{n-2}$ tels que $(a, (a_1, u_1)) \in \mu$, pour $1 \leq i \leq n-2$ on ait $(u_{i-1}, (a_i, u_i)) \in \mu$ et $(u_{n-2}, (a_{n-1}, a_n)) \in \mu$.

6.3. Remarque : Les espaces de corrélation négatifs sont des espaces cohérents munis d'une structure de monoïde pour l'opération $\parr$ (ou $\parr$-*monoïde*). Tandis que les espaces de corrélation positifs sont des espaces cohérents munis d'une structure de comonoïde pour l'opération $\otimes$ (ou $\otimes$-*comonoïde*).

Notation : Si $\mathcal{N}$ (resp. $\mathcal{P}$) est un espace de corrélation négatif (resp. positif), on notera également $\mathcal{N}$ (resp. $\mathcal{P}$) l'espace cohérent sous-jacent lorsque l'absence d'ambiguïté le permettra.

6.4. Définition (Constructions)
- Soit $\mathcal{N} = (\mathcal{N}, \eta, \mu)$ un espace de corrélation négatif, l'espace de corrélation positif $\mathcal{N}^\perp$ est défini par $\mathcal{N}^\perp = (\mathcal{N}^\perp, \varepsilon, \delta)$ avec :

$$\varepsilon = \eta$$
$$\delta = \{(z, (x, y)) \in |\mathcal{N}| \times (|\mathcal{N}| \times |\mathcal{N}|) \mid ((x, y), z) \in \mu\}$$

- Soit $\mathcal{P} = (\mathcal{P}, \varepsilon, \delta)$ un espace de corrélation positif, l'espace de corrélation négatif $\mathcal{P}^\perp$ est défini par $\mathcal{P}^\perp = (\mathcal{P}^\perp, \eta, \mu)$ avec :

$$\eta = \varepsilon$$
$$\mu = \{((x, y), z) \in (|\mathcal{N}| \times |\mathcal{N}|) \times |\mathcal{N}| \mid (z, (x, y)) \in \delta\}$$

- On note $\perp$ l'espace de corrélation négatif $(\perp, \{\star\}, \{((\star, \star), \star)\})$.
- On note $\top$ l'espace de corrélation négatif $(\top, \emptyset, \emptyset)$.
- Soient $\mathcal{N} = (\mathcal{N}, \eta_\mathcal{N}, \mu_\mathcal{N})$ et $\mathcal{M} = (\mathcal{M}, \eta_\mathcal{M}, \mu_\mathcal{M})$ deux espaces de corrélation négatifs, l'espace de corrélation négatif $\mathcal{N} \parr \mathcal{M}$ est défini par $\mathcal{N} \parr \mathcal{M} = (\mathcal{N} \parr \mathcal{M}, \eta_{\mathcal{N} \parr \mathcal{M}}, \mu_{\mathcal{N} \parr \mathcal{M}})$ avec :

$$\eta_{\mathcal{N} \parr \mathcal{M}} = \eta_\mathcal{N} \times \eta_\mathcal{M}$$
$$\mu_{\mathcal{N} \parr \mathcal{M}} = \{(((x, x'), (y, y')), (z, z')) \mid ((x, y), z) \in \mu_\mathcal{N} \wedge ((x', y'), z') \in \mu_\mathcal{M}\}$$

- Soient $\mathcal{N} = (\mathcal{N}, \eta_\mathcal{N}, \mu_\mathcal{N})$ et $\mathcal{M} = (\mathcal{M}, \eta_\mathcal{M}, \mu_\mathcal{M})$ deux espaces de corrélation négatifs, l'espace de corrélation négatif $\mathcal{N} \,\&\, \mathcal{M}$ est défini par $\mathcal{N} \,\&\, \mathcal{M} = (\mathcal{N} \,\&\, \mathcal{M}, \eta_{\mathcal{N} \& \mathcal{M}}, \mu_{\mathcal{N} \& \mathcal{M}})$ avec :

$$\eta_{\mathcal{N} \& \mathcal{M}} = \eta_\mathcal{N} + \eta_\mathcal{M}$$
$$\mu_{\mathcal{N} \& \mathcal{M}} = \{(((1, x), (1, y)), (1, z)) \mid ((x, y), z) \in \mu_\mathcal{N}\} \cup \{(((2, x), (2, y)), (2, z)) \mid ((x, y), z) \in \mu_\mathcal{M}\}$$

- Soit $\mathcal{A}$ un espace cohérent, l'espace de corrélation négatif $?\mathcal{A}$ est défini par $?\mathcal{A} = (?\mathcal{A}, \varepsilon, \delta)$ avec :

$$\varepsilon = \{[\,]\}$$
$$\delta = \{((x, y), x + y) \mid x, y, x + y \in |?\mathcal{A}|\}$$

- Les constructions positives sont obtenues par dualité : $1 = \perp^\perp$, $0 = \top^\perp$, $\mathcal{P} \otimes \mathcal{Q} = (\mathcal{P}^\perp \parr \mathcal{Q}^\perp)^\perp$, $\mathcal{P} \oplus \mathcal{Q} = (\mathcal{P}^\perp \,\&\, \mathcal{Q}^\perp)^\perp$ et $!\mathcal{P} = (?\mathcal{P}^\perp)^\perp$.

6.5. Remarque : Il est facile de vérifier que les constructions définies ci-dessus donnent lieu à des espaces de corrélation comme cela est affirmé. On a également $(\mathcal{N}^\perp)^\perp = \mathcal{N}$ et $(\mathcal{P}^\perp)^\perp = \mathcal{P}$.

On va montrer maintenant que les espaces de corrélation fournissent un modèle dénotationnel de LLP.

6.6. Définition (Interprétation de LLP)
Les formules de LLP sont naturellement interprétées par les espaces de corrélation associés (de la même polarité), les atomes (positifs) étant interprétés par des espaces de corrélation positifs arbitraires. Une preuve $\pi$ du séquent $\vdash A_1, ..., A_n$ est interprétée par la clique $[\![\pi]\!]_{coh}$ de l'espace $\mathcal{A}_1 \parr ... \parr \mathcal{A}_n$, obtenue comme dans la définition 2.6 pour les règles *ax*, *cut*, $\parr$, $\otimes$, &, $\oplus_i$, $\perp$, 1, $\top$ et $?d$ et pour les autres règles, si $\pi_1$ est la prémisse de la dernière règle de $\pi$ :

(!) $[\![\pi]\!]_{coh} = \{(y_1, ..., y_n, [z^1, ..., z^k]) \mid \exists (x^i_j)_{\substack{1\leq i\leq k \\ 1\leq j\leq n}} \forall 1 \leq j \leq n, y_j \leftarrow\!\!\!\shortmid \sum x^i_j \in \mathcal{A}_j \wedge \forall 1 \leq i \leq k, (x^i_1, ..., x^i_n, z^i) \in [\![\pi_1]\!]_{coh}\}$.

($?c$) $[\![\pi]\!]_{coh} = \{(x_1, ..., x_n, t) \mid \exists y \exists z\, (x_1, ..., x_n, y, z) \in [\![\pi_1]\!]_{coh} \wedge ((y, z), t) \in \mu_{\mathcal{N}}\}$

($?w$) $[\![\pi]\!]_{coh} = \{(x_1, ..., x_n, y) \mid (x_1, ..., x_n) \in [\![\pi_1]\!]_{coh} \wedge y \in \eta_{\mathcal{N}}\}$

6.7. Théorème (Sémantique dénotationnelle)
*Soit $\pi$ une preuve de* LLP *et $\pi'$ une preuve obtenue à partir de $\pi$ par élimination des coupures, on a $[\![\pi]\!]_{coh} = [\![\pi']\!]_{coh}$.*

Démonstration : Extension directe de la preuve de [Gir91a] pour LC. □

Ces résultats peuvent être étendus au second ordre sans problème grâce au travail de M. Quatrini [Qua96].

## 6.2 Catégories de contrôle

Les catégories de contrôle de P. Selinger sont une extension des *catégories cartésiennes fermées* permettant d'enrichir ces modèles du $\lambda$-calcul en des modèles du $\lambda\mu$-calcul.

Nous reprenons les définitions données dans [Sel01] pour les catégories de contrôle. Nous nous contentons d'énoncer les résultats généraux dont nous avons besoin à propos de ces catégories, on se référera à [Sel01] pour des preuves détaillées.

6.8. Définition (Foncteur binoïdal et catégorie binoïdale)
Soient $\mathbb{A}$, $\mathbb{B}$ et $\mathbb{C}$ trois catégories, un *foncteur binoïdal* $F$ de $\mathbb{A}$ et $\mathbb{B}$ dans $\mathbb{C}$ est la donnée pour tout objet $A$ de $\mathbb{A}$ d'un foncteur $F_A$ de $\mathbb{B}$ dans $\mathbb{C}$ et pour tout objet $B$ de $\mathbb{B}$ d'un foncteur $F_B$ de $\mathbb{A}$ dans $\mathbb{C}$ tels que : pour tout objet $A$ de $\mathbb{A}$ et tout objet $B$ de $\mathbb{B}$, on ait $F_A(B) = F_B(A)$.

On notera $F(A, B) = F_A(B) = F_B(A)$, $F(A, g) = F_A(g)$ et $F(f, B) = F_B(f)$ puisque ceci n'entraîne pas d'ambiguïté.

Une *catégorie binoïdale* $\mathbb{A}$ est une catégorie munie d'un foncteur binoïdal $\parr$ de $\mathbb{A}$ et $\mathbb{A}$ dans $\mathbb{A}$ (pour lequel on utilisera la notation infixe).

6.9. Remarque : Si $F$ est un foncteur binoïdal, $f : A \to A'$ et $g : B \to B'$, la notation $F(f, g)$ n'est a priori pas définie parce qu'en général $F(f, B') \circ F(A, g) \neq F(A', g) \circ F(f, B)$, ce qui rendrait la notation ambiguë.

6.10. Définition (Centre)
Soit $\mathbb{A}$ une catégorie binoïdale, le morphisme $f : A \to A'$ est *central*, noté $f : A \xrightarrow{\bullet} A'$, si pour tout morphisme $g : B \to B'$, $(f \parr B') \circ (A \parr g) = (A' \parr g) \circ (f \parr B)$.

Si $f$ ou $g$ est central on notera $f \parr g = (f \parr B') \circ (A \parr g) = (A' \parr g) \circ (f \parr B)$.

Le *centre* $\mathbb{A}^\bullet$ de $\mathbb{A}$ est la catégorie ayant les mêmes objets que $\mathbb{A}$ et comme morphismes uniquement les morphismes centraux.

6.11. Définition (Catégorie prémonoïdale)
Une catégorie binoïdale $\mathbb{A}$ est *prémonoïdale* si le foncteur binoïdal $\parr$ est associatif et possède un élément neutre $\perp$ (les isomorphismes correspondants $a_{A,B,C} : (A \parr B) \parr C \to A \parr (B \parr C)$, $l_A : A \to A \parr \perp$ et $r_A : A \to \perp \parr A$ devant être centraux).

Une catégorie prémonoïdale est *symétrique* si, de plus, $\parr$ est commutatif (les morphismes correspondants $c_{A,B} : A \parr B \to B \parr A$ étant centraux).

6.12. Remarque : Le centre $\mathbb{A}^\bullet$ d'une catégorie prémonoïdale $\mathbb{A}$ est une catégorie monoïdale.

6.13. Définition (Monoïde)
Soit $\mathbb{A}$ une catégorie prémonoïdale symétrique et $A$ un objet de $\mathbb{A}$, le triplet $(A, \eta, \mu)$ avec les morphismes centraux $\eta : \perp \overset{\bullet}{\to} A$ et $\mu : A \parr A \overset{\bullet}{\to} A$ est un *monoïde* si :

(neutralité) $\mu \circ (A \parr \eta) = {l_A}^{-1}$ et $\mu \circ (\eta \parr A) = {r_A}^{-1}$

(associativité) $\mu \circ (\mu \parr A) = \mu \circ (A \parr \mu) \circ a_{A,A,A}$

(commutativité) $\mu \circ c_{A,A} = \mu$

Notations : Soit $\mathbb{A}$ une *catégorie cartésienne fermée* (CCC), on note :
- $\&$ le produit cartésien,
- $\top$ l'objet terminal,
- $<f, g> : C \to A \& B$ la paire des morphismes $f : C \to A$ et $g : C \to B$,
- $\pi_1 : A \& B \to A$ et $\pi_2 : A \& B \to B$ les deux projections,
- $t_A : A \to \top$ le morphisme d'objet terminal,
- $ev : B^A \& A \to B$ le morphisme d'évaluation,
- $f^* : \top \to B^A$ le curryfié de $f : A \to B$.

6.14. Définition (Distributivité)
Soit $\mathbb{A}$ une catégorie cartésienne et prémonoïdale symétrique dont tout objet $A$ est muni d'une structure de monoïde $(A, \eta_A, \mu_A)$, on dit que $\mathbb{A}$ est *distributive* si le foncteur binoïdal $\parr$ est distributif sur le produit cartésien $\&$. C'est-à-dire que l'on a :
- $d_{A,B,C} = <A \parr \pi_1, A \parr \pi_2> : A \parr (B \& C) \overset{\bullet}{\to} (A \parr B) \& (A \parr C)$ est un isomorphisme,
- $ab_A = t_{A \parr \top} : A \parr \top \overset{\bullet}{\to} \top$ est un isomorphisme.

On note $d'_{A,B,C} = l_A \& (B \parr C); (A \parr \eta_C) \& (B \parr C); {d_{C,A,B}}^{-1} : A \& (B \parr C) \overset{\bullet}{\to} (A \& B) \parr C$ et on l'appellera morphisme de *faible distributivité*.

6.15. Définition (Catégorie de contrôle)
Une *catégorie de contrôle* est une catégorie cartésienne fermée et prémonoïdale symétrique telle que :
- Tout objet $A$ est muni d'une structure de monoïde $(A, \eta_A, \mu_A)$.
- Elle est distributive.
- Le morphisme $s_{A,B,C} : B^A \parr C \to (B \parr C)^A$ obtenu en curryfiant :

$$(B^A \parr C) \& A \xrightarrow{d'_{A,B^A,C}} (B^A \& A) \parr C \xrightarrow{ev \parr C} (B \parr C)$$

est un isomorphisme naturel en $A$, $B$ et $C$.

– Les diagrammes suivants commutent :

$$\begin{array}{ccccc}
B^A \parr C^D & & \xrightarrow{s_{D,C,B^A}} & & (B^A \parr C)^D \\
{\scriptstyle s_{A,B,C^D}}\Big\downarrow & & & & \Big\downarrow{\scriptstyle s^D_{A,B,C}} \\
(B \parr C^D)^A & \xrightarrow{s^A_{D,C,B}} & ((B \parr C)^D)^A & \xrightarrow{ccc} & ((B \parr C)^A)^D
\end{array}$$

$$\begin{array}{ccc}
B^A \parr B^A & \xrightarrow{\mu} & B^A \\
{\scriptstyle s_{A,B,B^A}}\Big\downarrow & & \Big\uparrow{\scriptstyle \mu^{<id_A,id_A>}} \\
(B^A \parr B)^A & \xrightarrow{s^A_{A,B,B}} & (B \parr B)^{A \& A}
\end{array}$$

et $\eta_{B^A} = \eta_B^{t_A} : \perp \to B^A$.

6.16. Définition (Catégorie de co-contrôle)
La catégorie $\mathbb{A}$ est une *catégorie de co-contrôle* si sa catégorie opposée $\mathbb{A}^{\mathrm{op}}$ est une catégorie de contrôle. On note $\otimes$ le produit tensoriel (1 son élément neutre) et $\oplus$ le co-produit (0 son élément neutre).

6.17. Lemme (Propriétés des morphismes centraux)
*Soit $\mathbb{A}$ une catégorie de contrôle et $f : A \overset{\bullet}{\to} B$ un morphisme central, $f$ est* effaçable, *i.e.* $f \circ \eta_A = \eta_B$ *et $f$ est* copiable, *i.e.* $f \circ \mu_A = \mu_B \circ (f \parr f)$.

6.18. Lemme (Définition et propriétés de $p_{A,B,C}$)
*Le morphisme $p_{A,B,C} : B^A \to (C \parr B)^{C \parr A}$ défini par curryfication de :*

$$B^A \,\&\, (C \parr A) \xrightarrow{\eta \& (C \parr A)} (C \parr B^A) \,\&\, (C \parr A) \xrightarrow{{d_{C,B^A,A}}^{-1}} C \parr (B^A \,\&\, A) \xrightarrow{C \parr ev} C \parr B$$

*est central et vérifie les conditions de cohérence suivantes :*

$$\begin{array}{ccc}
B^A \parr C & \xrightarrow{p_{A,B,D \parr C}} & (D \parr B)^{D \parr A} \parr C \\
{\scriptstyle s_{A,B,C}}\Big\downarrow & & \Big\downarrow{\scriptstyle s_{D \parr A, D \parr B, C}} \\
(B \parr C)^A & \xrightarrow{p_{A,B \parr C,D}} & (D \parr B \parr C)^{D \parr A}
\end{array}$$

*et* ${id_A}^*; p_{A,A,B} = {id_{B \parr A}}^*$.

6.19. Lemme (Réversibilité)
*Soit $\mathbb{A}$ une catégorie de contrôle,* $\mathbb{A}(\top, A \parr B) \simeq \mathbb{A}^\bullet(\perp^A, B)$.

Démonstration : Sans donner plus de détails, si $f : \top \to A \parr B$ on obtient le morphisme $\perp^A \overset{\bullet}{\to} B$ par :

$$\perp^A \xrightarrow{p_{A,\perp,B}} (B \parr \perp)^{B \parr A} \xrightarrow{{l_B}^{-1\,B \parr A}} B^{B \parr A} \xrightarrow{B^f} B^\top \xrightarrow{ccc} B$$

le reste de la preuve se trouve dans [Sel01]. □

Nous allons maintenant montrer que les catégories de contrôle et de co-contrôle fournissent des modèles dénotationnels de LLP.

6.20. Définition (Interprétation négative de LLP)
Soit $\mathbb{A}$ une catégorie de contrôle, on suppose fixée une interprétation des atomes négatifs $X^\perp$ par des objets $\llbracket X^\perp \rrbracket^-_{cc}$ de $\mathbb{A}$, l'interprétation des formules négatives est donnée de la manière suivante :

$$\begin{array}{ccc} N \parr M & \leadsto & \llbracket N \rrbracket^-_{cc} \parr \llbracket M \rrbracket^-_{cc} \\ N \mathbin{\&} M & \leadsto & \llbracket N \rrbracket^-_{cc} \mathbin{\&} \llbracket M \rrbracket^-_{cc} \\ \perp & \leadsto & \perp \\ \top & \leadsto & \top \\ ?N^\perp & \leadsto & \perp^{\llbracket N \rrbracket^-_{cc}} \end{array}$$

Une preuve $\pi$ est interprétée par un morphisme $\llbracket \pi \rrbracket^-_{cc}$ de la catégorie avec deux cas selon la forme du séquent conclusion :

$$\begin{array}{lcl} \vdash N_1, \ldots, N_n & \leadsto & \llbracket \pi \rrbracket^-_{cc} : \top \to \llbracket N_1 \rrbracket^-_{cc} \parr \ldots \parr \llbracket N_n \rrbracket^-_{cc} \\ \vdash N_1, \ldots, N_n, P & \leadsto & \llbracket \pi \rrbracket^-_{cc} : \llbracket P^\perp \rrbracket^-_{cc} \overset{\bullet}{\to} \llbracket N_1 \rrbracket^-_{cc} \parr \ldots \parr \llbracket N_n \rrbracket^-_{cc} \end{array}$$

en prenant $\perp$ à droite si $n = 0$.

On définit $\llbracket \pi \rrbracket^-_{cc}$ par récurrence sur la taille de $\pi$ selon sa dernière règle. Pour simplifier l'écriture, et puisqu'il n'y a pas de confusion possible, on omettra les $\llbracket . \rrbracket^-_{cc}$ sur les formules. Afin de factoriser des cas similaires, on notera $f : \Pi \overset{\bullet}{\to} A$ un morphisme $f$ pouvant être soit $f : \top \to A$ soit $f : B \overset{\bullet}{\to} A$ (en accord avec les règles de LLP telles qu'elles sont écrites à la section 10.1.3).

($ax$) $\llbracket \pi \rrbracket^-_{cc} = id_{\llbracket A \rrbracket^-_{cc}} : A \overset{\bullet}{\to} A$

($cut$) Si $A$ est la formule coupée, on obtient par hypothèse de récurrence un morphisme $f : \Pi \overset{\bullet}{\to} C \parr A$ et un morphisme $g : A \overset{\bullet}{\to} B$. On a $\llbracket \pi \rrbracket^-_{cc} = f; (C \parr g) : \Pi \overset{\bullet}{\to} C \parr B$.

($\parr$) Par hypothèse de récurrence, on obtient un morphisme $f : \Pi \overset{\bullet}{\to} A \parr B \parr C$ et on construit $\llbracket \pi \rrbracket^-_{cc} : \Pi \overset{\bullet}{\to} A \parr (B \parr C)$ avec $a_{A,B,C}$.

($\otimes$) Par hypothèse de récurrence, on obtient deux morphismes $f : A \overset{\bullet}{\to} C$ et $g : B \overset{\bullet}{\to} D$ et on a $\llbracket \pi \rrbracket^-_{cc} = f \parr g : A \parr B \overset{\bullet}{\to} C \parr D$.

($\&$) Par hypothèse de récurrence, on obtient deux morphismes $f : \Pi \overset{\bullet}{\to} A \parr B$ et $g : \Pi \overset{\bullet}{\to} A \parr C$ et on a $\llbracket \pi \rrbracket^-_{cc} = <f, g>; {d_{A,B,C}}^{-1} : \Pi \overset{\bullet}{\to} A \parr (B \mathbin{\&} C)$.

($\oplus_i$) Par hypothèse de récurrence, on obtient un morphisme $f : A_i \overset{\bullet}{\to} B$ et on construit $\llbracket \pi \rrbracket^-_{cc} = \pi_i; f : A_1 \mathbin{\&} A_2 \overset{\bullet}{\to} B$.

($\perp$) Par hypothèse de récurrence, on obtient un morphisme $f : \Pi \overset{\bullet}{\to} A$ et on construit $\llbracket \pi \rrbracket^-_{cc} = f; l_A : \Pi \overset{\bullet}{\to} A \parr \perp$.

(1) $\llbracket \pi \rrbracket^-_{cc} = id_\perp : \perp \overset{\bullet}{\to} \perp$

($\top$) $\llbracket \pi \rrbracket^-_{cc} = t_\Pi; {ab_A}^{-1} : \Pi \overset{\bullet}{\to} A \parr \top$

(!) Par hypothèse de récurrence, on obtient un morphisme $f : \top \to A \parr B$ et, en appliquant le lemme 6.19, on obtient le morphisme $\llbracket \pi \rrbracket^-_{cc} : \perp^A \overset{\bullet}{\to} B$.

($?d$) Par hypothèse de récurrence, on obtient un morphisme $f : A \overset{\bullet}{\to} B$, par curryfication on obtient un morphisme $f^* : \top \to B^A$. On construit $\llbracket \pi \rrbracket^-_{cc} = g; l_B; {s_{A,\perp,B}}^{-1} : \top \to \perp^A \parr B$.

($?c$) Par hypothèse de récurrence, on obtient un morphisme $f : \Pi \overset{\bullet}{\to} A \parr B \parr B$ et on construit $\llbracket \pi \rrbracket^-_{cc} = f; (A \parr \mu_B) : \Pi \overset{\bullet}{\to} A \parr B$.

($?w$) Par hypothèse de récurrence, on obtient un morphisme $f : \Pi \overset{\bullet}{\to} A$ et on construit $\llbracket \pi \rrbracket^-_{cc} = f; l_A; (A \parr \eta_B) : \Pi \overset{\bullet}{\to} A \parr B$.

6.21. Théorème (Sémantique catégorique négative de LLP)
*Soit $\mathbb{A}$ une catégorie de contrôle, si la preuve $\pi$ se réduit en $\pi'$ alors* $[\![\pi]\!]^-_{cc} = [\![\pi']\!]^-_{cc}$.

Démonstration : On montre que l'interprétation est préservée par élimination des coupures. Le cas le plus important (et le plus complexe) est celui d'une coupure $!/?d$ :

Soient $f : \top \to A \parr B$ et $g : A \overset{\bullet}{\to} C$, on note $!f : \perp^A \overset{\bullet}{\to} B$ le morphisme obtenu à partir de $f$ dans le lemme 6.19 et $?g = g^*; l^A_C; {s_{A,\perp,C}}^{-1} : \top \to C \parr \perp^A$, on doit montrer que $?g; C \parr !f = f; g \parr B : \top \to C \parr B$.

1. On commence par réécrire $?g$

$$\begin{tikzcd}
\top \arrow[r, "g^*"] \arrow[d, "{l_A}^*"'] & C^A \arrow[d, "l^A_C"] \\
(A \parr \perp)^A \arrow[r, "(g \parr \perp)^A"] \arrow[d, "{s_{A,\perp,A}}^{-1}"'] & (C \parr \perp)^A \arrow[d, "{s_{A,\perp,C}}^{-1}"] \\
A \parr \perp^A \arrow[r, "g \parr \perp^A"] & C \parr \perp^A
\end{tikzcd}$$

le premier carré commute par propriétés des CCCs et le second par naturalité en $C$ de $s_{A,B,C}$.

2. On s'intéresse désormais à $g \parr \perp^A; C \parr !f$

$$\begin{tikzcd}
A \parr \perp^A \arrow[r, "g \parr \perp^A"] \arrow[d, "A \parr !f"'] & C \parr \perp^A \arrow[d, "C \parr !f"] \\
A \parr B \arrow[r, "g \parr B"] & C \parr B
\end{tikzcd}$$

ce diagramme commute par centralité de $!f$ et/ou $g$.

3. En composant les deux diagrammes précédents, on a $?g; C \parr !f = {l_A}^*; {s_{A,\perp,A}}^{-1}; A \parr !f; g \parr B$. Il nous reste donc à montrer que ${l_A}^*; {s_{A,\perp,A}}^{-1}; A \parr !f = f : \top \to A \parr B$, pour cela on utilise la définition de $!f$ donnée dans la preuve du lemme 6.19 :

$$\begin{tikzcd}
\top \arrow[d, "{l_A}^*"'] & & \\
(A \parr \perp)^A \arrow[r, "{s_{A,\perp,A}}^{-1}"] \arrow[d, "p_{A,A \parr \perp,B}"'] & A \parr \perp^A \arrow[d, "A \parr p_{A,\perp,B}"] & \\
(A \parr B \parr \perp)^{A \parr B} \arrow[r, "{s_{A,B \parr \perp,A}}^{-1}"] \arrow[d, "{l_{A \parr B}}^{-1}{}^{A \parr B}"'] & A \parr (B \parr \perp)^{A \parr B} \arrow[d, "A \parr {l_B}^{-1}{}^{A \parr B}"] & \\
(A \parr B)^{A \parr B} \arrow[r, "{s_{A \parr B,B,A}}^{-1}"] \arrow[d, "(A \parr B)^f"'] & A \parr B^{A \parr B} \arrow[d, "A \parr B^f"] & \\
(A \parr B)^\top \arrow[r, "{s_{\top,B,A}}^{-1}"] & A \parr B^\top \arrow[r, "ccc"] & A \parr B
\end{tikzcd}$$

le premier carré commute par la première condition de cohérence du lemme 6.18 et les deux autres par naturalité de $s_{A,B,C}$. En utilisant enfin la deuxième condition de cohérence du lemme 6.18 et les propriétés de CCC, on conclut que le morphisme $\top \to A \parr B$ obtenu est $f$.

Les autres cas nécessitant des applications nettement plus immédiates des définitions, ils sont laissés au lecteur. □

6.22. Définition (Interprétation positive de LLP)
Soit $\mathbb{A}$ une catégorie de co-contrôle, on suppose fixée une interprétation des atomes positifs $X$ par des objets $[\![X]\!]^+_{cc}$ de $\mathbb{A}$, l'interprétation des formules positives est donnée de la manière suivante :

$$\begin{array}{ccc} P \otimes Q & \rightsquigarrow & [\![P]\!]^+_{cc} \otimes [\![Q]\!]^+_{cc} \\ P \oplus Q & \rightsquigarrow & [\![P]\!]^+_{cc} \oplus [\![Q]\!]^+_{cc} \\ 1 & \rightsquigarrow & 1 \\ 0 & \rightsquigarrow & 0 \\ !P^\perp & \rightsquigarrow & 1^{[\![P]\!]^+_{cc}} \end{array}$$

Une preuve $\pi$ est interprétée par un morphisme $[\![\pi]\!]^+_{cc}$ de la catégorie avec deux cas selon la forme du séquent conclusion :

$$\begin{array}{lll} \vdash N_1, \ldots, N_n & \rightsquigarrow & [\![\pi]\!]^+_{cc} : [\![N_1^\perp]\!]^+_{cc} \otimes \cdots \otimes [\![N_n^\perp]\!]^+_{cc} \to 0 \\ \vdash N_1, \ldots, N_n, P & \rightsquigarrow & [\![\pi]\!]^+_{cc} : [\![N_1^\perp]\!]^+_{cc} \otimes \cdots \otimes [\![N_n^\perp]\!]^+_{cc} \stackrel{\bullet}{\to} [\![P]\!]^+_{cc} \end{array}$$

en prenant 1 à gauche si $n = 0$.

La définition de l'interprétation des preuves ainsi que la démonstration de la proposition suivante se déduisent aisément du cas des catégories de contrôle puisqu'on ne fait que passer à la structure opposée.

6.23. Proposition (Sémantique catégorique positive de LLP)
*Soit $\mathbb{A}$ une catégorie de co-contrôle, si la preuve $\pi$ se réduit en $\pi'$ alors $[\![\pi]\!]^+_{cc} = [\![\pi']\!]^+_{cc}$.*

Puisque le $\lambda\mu$-calcul par nom (resp. par valeur) est un langage interne pour les catégories de contrôle (resp. co-contrôle), on a ainsi établi sémantiquement les liens qui unissent le $\lambda\mu$-calcul et LLP. On verra, au chapitre 11, la version syntaxique de ces résultats à travers les traductions du $\lambda\mu$-calcul dans LLP. Il en découle en particulier que LLP est un langage interne pour les catégories de contrôle et de co-contrôle.

# Chapitre 7

# Jeux polarisés

Un des défauts majeurs des jeux intuitionnistes est l'absence de dualité ou de négation qui nécessiteraient une parfaite symétrie entre Joueur et Opposant. Nous allons montrer qu'il est possible d'étendre ces jeux en des jeux polarisés pour obtenir cette dualité. Un des ingrédients principaux est l'utilisation d'opérateurs de décalage $\downarrow$ et $\uparrow$, déjà introduits dans [Lam95, Gir01]. Ainsi les jeux polarisés possèdent un grain suffisamment fin pour permettre de clarifier différentes propriétés des jeux intuitionnistes.

Bien que la Logique Linéaire ait servi de guide dans la définition des sémantiques de jeux, et bien que la dualité soit au cœur de LL, ceci n'a pas permis de définir de bonne notion de jeux avec une réelle dualité. Le lien avec LLP est donc essentiel. Mieux encore, le résultat de définissabilité que l'on obtient pour les jeux polarisés est un résultat très fort de sémantique dénotationnelle que l'on ne sait obtenir que pour des systèmes particuliers et en tout cas pas LL tout entier.

Les jeux polarisés s'avèrent être un outil particulièrement riche et précis pour l'analyse de LLP et donc des systèmes classiques.

Nous présentons ici deux variantes possibles pour l'interprétation des connecteurs exponentiels qui correspondent aux deux principales traditions de jeux intuitionnistes : les jeux HO et les jeux AJM.

## 7.1 Jeux HO polarisés

On étend certaines définitions du chapitre 3 pour obtenir une véritable dualité entre les deux joueurs $O$ et $J$. Il s'agit principalement d'autoriser $J$ à commencer les parties.

7.1. Définition (Arène HO polarisée)
Une *arène polarisée* $A$ est un quadruplet $(\pi_A, M_A, \lambda_A, \vdash_A)$ où :

- $\pi_A$ est soit $O$ (ou de manière équivalente $-$) soit $J$ (ou $+$), appelé *polarité* de l'arène,
- $M_A$ est un ensemble appelé ensemble de *coups*,
- $\lambda_A$ est une fonction de $M_A$ dans $\{O, J\}$,
- $\vdash_A$ est la donnée d'un sous-ensemble $M_A^i$ de $M_A$ appelé ensemble des *coups initiaux* (on note $\vdash_A m$ si $m$ est dans cet ensemble et on note $M_A^{ni}$ l'ensemble des coups non initiaux) et d'une relation binaire sur $M_A$ appelée *relation de validation* (on note $m \vdash_A n$ si $m$ et $n$ sont en relation). $\vdash_A$ doit vérifier :
  - Si $\vdash_A m$ alors $\lambda_A(m) = \pi_A$ et pour tout coup $n \in M_A$, $n \nvdash_A m$.
  - Si $m \vdash_A n$ alors $\lambda_A(m) \neq \lambda_A(n)$

7.2. Remarque : Une arène polarisée négative est exactement une arène au sens de la définition 3.1. En particulier toutes les constructions que l'on a vues au chapitre 3 pour les arènes ($+$, $\Rightarrow$, ...) s'appliquent aux arènes polarisées négatives et de même pour les jeux ($\odot$, $\multimap$, $\&$, $\sharp$, ...).

7.3. Définition (Vue)
Soit $s$ une suite justifiée, on définit la *vue joueur* de $s$, notée $\ulcorner s \urcorner$, qui est une sous-suite justifiée, par :

- $\ulcorner \varepsilon \urcorner = \varepsilon$
- $\ulcorner sm^J \urcorner = \ulcorner s \urcorner m^J$
- $\ulcorner sm^O \urcorner = m^O$ si $m$ est initial
- $\ulcorner smtn^O \urcorner = \ulcorner s \urcorner mn^O$ si $n$ est justifié par $m$

La *vue opposant* $\llcorner s \lrcorner$ est définie de manière symétrique.

7.4. Définition (Position légale)
Une *position légale* est une suite justifiée $s$ telle que :

- Si $tmn \leq s$ alors $\lambda(m) \neq \lambda(n)$.
- Si $tm^J \leq s$ et $m$ n'est pas initial alors $m$ pointe dans $\ulcorner t \urcorner$.
- Si $tm^O \leq s$ et $m$ n'est pas initial alors $m$ pointe dans $\llcorner t \lrcorner$.

On note $L_A$ l'ensemble des positions légales de l'arène $A$.

Puisque si $n$ est justifié par $m$ dans $s$ et $n$ est dans $\ulcorner s \urcorner$ alors $m$ est dans $\ulcorner s \urcorner$, on conserve dans $\ulcorner s \urcorner$ les mêmes pointeurs que dans $s$.

7.5. Définition (Constructions d'arènes polarisées)
Décalage. Soit $A$ une arène polarisée, on note $\updownarrow A$ l'*arène décalage* de polarité $\overline{\pi_A}$ (*i.e.* de polarité opposée à $\pi_A$) définie par :

- $M_{\updownarrow A} = \{\star\} + M_A$
- $\lambda_{\updownarrow A}(\star) = \overline{\pi_A}$
- $\lambda_{\updownarrow A}(m) = \lambda_A(m)$ si $m \in M_A$
- $\vdash_{\updownarrow A} \star$
- $\star \vdash_{\updownarrow A} m$ *ssi* $\vdash_A m$
- $m \vdash_{\updownarrow A} n$ *ssi* $m \vdash_A n$ pour $m, n \in M_A$

Produit. Soient $A$ et $B$ deux arènes de même polarité, on note $A \times B$ l'*arène produit* définie par :

- $M_{A\times B} = M_A^i \times M_B^i + M_A^{ni} + M_B^{ni}$ sauf si $M_A = \emptyset$ (ou $M_B = \emptyset$) auquel cas $M_{A\times B} = \emptyset$ (sinon des coups non initiaux dans $A$ et dans $B$ deviendraient initiaux dans $A \times B$).
- $\lambda_{A\times B}(m_1, m_2) = \pi_A = \pi_B$ avec $m_1$ initial dans $A$ et $m_2$ initial dans $B$
- $\lambda_{A\times B}(m) = [\lambda_A, \lambda_B](m)$ si $m$ n'est pas initial
- $\vdash_{A\times B} (m_1, m_2)$ avec $m_1$ initial dans $A$ et $m_2$ initial dans $B$
- $(m_1, m_2) \vdash_{A\times B} n$ si $m_1$ et $n$ sont dans $A$ et $m_1 \vdash_A n$ ou si $m_2$ et $n$ sont dans $B$ et $m_2 \vdash_B n$
- $m \vdash_{A\times B} n$ si $m$ et $n$ sont dans $A$ et $m \vdash_A n$ ou si $m$ et $n$ sont dans $B$ et $m \vdash_B n$ lorsque $m$ n'est pas initial

Si $s$ est une position légale *bien ouverte* de $A \times B$, on note $s \upharpoonright_A$ la sous-suite de $s$ ne contenant que les coups dans $A$, c'est une position légale de $A$ (et de même pour $B$).

7.6. Remarque : On ne considérera que les positions légales bien ouvertes dans $A \times B$, cette contrainte étant cohérente avec le fait qu'il faut se restreindre aux jeux bien ouverts pour construire la CCC de jeux HO de la proposition 3.25. On pourrait considérer les suites légales quelconques de l'arène $A \times B$ mais cela compliquerait nettement les choses puisqu'en particulier $s \upharpoonright_A$ n'est alors plus nécessairement une position légale de $A$, l'alternance des joueurs peut être violée. Dans le cas particulier bien ouvert, la construction $\times$ correspond à celle introduite par Honda et Yoshida [HY99].

7.7. Définition (Jeu HO polarisé)
Un *jeu polarisé* $A$ est un quintuplet $(\pi_A, M_A, \lambda_A, \vdash_A, \mathcal{P}_A)$ où $(\pi_A, M_A, \lambda_A, \vdash_A)$ est une arène polarisée et $\mathcal{P}_A$, appelé l'ensemble des *parties* de $A$, est un ensemble de positions légales de cette arène non vide et clos par préfixe qui vérifie que :

(i) Si $I$ est un ensemble de coups initiaux de $s \in \mathcal{P}_A$ alors $s \upharpoonright_I \in \mathcal{P}_A$.

7.8. Remarque : Si $\mathcal{P}_A \neq \{\varepsilon\}$, l'information donnée par $\pi_A$ est redondante.

7.9. Définition (Constructions de jeux polarisés)

($(.)^\perp$) $A^\perp = (\overline{\pi_A}, M_A, \overline{\lambda_A}, \vdash_A, \mathcal{P}_A)$.

($\uparrow$) Si $P$ est positif, $\uparrow P$ (*décalage* de $P$) est le jeu négatif dont l'arène est $\updownarrow P$ et dont l'ensemble de parties est $\mathcal{P}_{\uparrow P} = \star.\mathcal{P}_P \cup \{\varepsilon\}$.

($\perp$) On note $\perp$ le jeu ayant un coup $\star$ et les deux parties possibles : $(\{\star\}, \lambda_\perp(\star) = O, \vdash_\perp \star, \{\varepsilon, \star\})$.

($⅋$) Si $N$ et $M$ sont négatifs et bien ouverts, le jeu négatif bien ouvert $N ⅋ M$ a comme arène $N \times M$ et comme ensemble de parties $\mathcal{P}_{N⅋M} = \{s \in L_{N\times M} \mid s \upharpoonright_N \in \mathcal{P}_N \wedge s \upharpoonright_M \in \mathcal{P}_M\}$.

($?$) Si $P$ est un jeu positif, on note $?P$ le jeu $\uparrow\flat P$ où $\flat P = (\sharp P^\perp)^\perp$ (cette idée de décomposition des constructions exponentielles est initialement due à J.-Y. Girard [Gir01]).

($\multimap$) Si $P$ est un jeu positif et $N$ un jeu négatif, on note $P \multimap N$ le jeu $P^\perp ⅋ N$.

On définit également, par dualité, les constructions donnant lieu à des jeux positifs : $0 = \top^\perp$, $1 = \perp^\perp$, $P \otimes Q = (P^\perp ⅋ Q^\perp)^\perp$, $P \oplus Q = (P^\perp \& Q^\perp)^\perp$, $\downarrow N = (\uparrow N^\perp)^\perp$ et $!N = \downarrow\sharp N$ où $P$ et $Q$ sont positifs et $N$ est négatif.

7.10. Remarque : Soit $smn$ une partie de $N ⅋ M$, si $m \in M_N$ (resp. $m \in M_M$) et $n \in M_M$ (resp. $n \in M_N$), on a $\lambda_{N⅋M}(m) = 0$ et $\lambda_{N⅋M}(n) = J$, c'est la *condition de switching* de $⅋$.

7.11. Proposition (Propriétés structurelles)

- *$⅋$ est a-commutatif et a-associatif pour les jeux bien ouverts :*

$$A ⅋ B \simeq_a B ⅋ A \qquad \textit{si } A \textit{ et } B \textit{ sont bien ouverts}$$
$$A ⅋ (B ⅋ C) \simeq_a (A ⅋ B) ⅋ C \qquad \textit{si } A, B \textit{ et } C \textit{ sont bien ouverts}$$

- *$\perp$ est a-neutre pour $⅋$ par rapport aux jeux bien ouverts :*

$$A ⅋ \perp \simeq_a A \qquad \textit{si } A \textit{ est bien ouvert}$$

- *$\top$ est a-absorbant pour $⅋$ :*

$$A ⅋ \top \simeq_a \top$$

- *$⅋$ est p-distributif par rapport à & pour les jeux bien ouverts :*

$$A ⅋ (B \& C) \simeq_p (A ⅋ B) \& (A ⅋ C) \qquad \textit{si } A, B \textit{ et } C \textit{ sont bien ouverts}$$

- *autres propriétés :*

$$
\begin{array}{rll}
\downarrow(A \odot B) \simeq_a \downarrow A \otimes \downarrow B & & \\
!(A \,\&\, B) \simeq_a\, !A \otimes !B & & \\
A \multimap\!\!\!\!\!\triangleright\ B \simeq_a \uparrow A^{\perp} \parr B & & \textit{si } B \textit{ est bien ouvert} \\
\sharp A \multimap\!\!\!\!\!\triangleright\ B \simeq_a\, !A \multimap B & & \textit{si } B \textit{ est bien ouvert} \\
\downarrow\top = 1 & & \\
!\top = 1 & &
\end{array}
$$

Démonstration :

- $a$-commutativité et $a$-associativité de $\parr$ sont immédiates, la bonne ouverture des jeux étant nécessaire ici.
- L'arène du jeu $A \parr \perp$ a pour coups initiaux les paires $(a, \star)$ avec $a$ initial dans $A$ et pour coups non initiaux les coups non initiaux de $A$. On considère la fonction $f$ de $M_{A \parr \perp}$ dans $M_A$ qui associe $a$ à $(a, \star)$ pour $a$ initial et $a$ à $a$ pour $a$ non initial. Il est clair que $f$ est un $a$-isomorphisme de $A \parr \perp$ dans $A$.
- Le jeu $A \parr \top$ a un ensemble de coups vide, il est donc $a$-isomorphe à $\top$.
- Les coups initiaux de l'arène $(A \parr B) \,\&\, (A \parr C)$ sont les coups $(1, (a, b))$ et $(2, (a, c))$ avec $a$, $b$ et $c$ initiaux dans $A$, $B$ et $C$. Les coups non initiaux de cette arène sont les coups $(1, a)$, $(1, b)$, $(2, a)$ et $(2, c)$ avec $a$, $b$ et $c$ non initiaux dans $A$, $B$ et $C$. Soit $f$ la fonction de $M_{(A \parr B) \& (A \parr C)}$ dans $M_{A \parr (B \& C)}$ définie par :

$$
\begin{array}{ccc}
(1, (a, b)) & \mapsto & (a, (1, b)) \\
(2, (a, c)) & \mapsto & (a, (2, c)) \\
(1, a) & \mapsto & a \\
(1, b) & \mapsto & (1, b) \\
(2, a) & \mapsto & a \\
(2, c) & \mapsto & (2, c)
\end{array}
$$

  $f$ n'est pas un $a$-isomorphisme puisque qu'elle n'est pas injective mais la fonction induite sur les ensemble de parties est un $p$-isomorphisme. Pour cela il faut essentiellement remarquer que deux coups $(1, a)$ et $(2, a')$ (avec $a$ et $a'$ coups de $A$) ne peuvent pas être utilisés dans une même partie de $(A \parr B) \,\&\, (A \parr C)$.
- L'unique coup initial de $\downarrow(A \odot B)$ est $\star$ et l'unique coup initial de $\downarrow A \otimes \downarrow B$ est la paire $(\star, \star)$. Les coups non initiaux de $\downarrow(A \odot B)$ et de $\downarrow A \otimes \downarrow B$ sont les coups de $A$ et de $B$. On considère la fonction $f$ de $M_{\downarrow(A \odot B)}$ dans $M_{\downarrow A \otimes \downarrow B}$ définie par $f(\star) = (\star, \star)$, $f(a) = a$ et $f(b) = b$ pour les coups de $A$ et de $B$. On doit montrer que l'image par $f$ et par $f^{-1}$ d'une partie est une partie. Soit $s$ une partie de $\downarrow(A \odot B)$, si c'est la partie vide le résultat est immédiat et sinon $s = \star.t$ avec $t \upharpoonright_A \in \mathcal{P}_A$, donc $f(s) \upharpoonright_{\downarrow A} = \star.(f(t) \upharpoonright_A) = \star.(t \upharpoonright_A)$ est une partie de $\downarrow A$ et de même pour la projection sur $\downarrow B$ d'où $f(s)$ est une partie de $\downarrow A \otimes \downarrow B$. De la même manière, on montre que si $s$ est une partie de $\downarrow A \otimes \downarrow B$ alors $f^{-1}(s)$ est une partie de $\downarrow(A \odot B)$.
- Il s'agit de la composition de l'isomorphisme précédent et de celui de la proposition 3.22 :

$$
!(A \,\&\, B) = \downarrow\sharp(A \,\&\, B) \simeq_a \downarrow(\sharp A \odot \sharp B) \simeq_a \downarrow\sharp A \otimes \downarrow\sharp B = !A \otimes !B
$$

- Les coups initiaux de $A \multimap\!\!\!\!\!\triangleright\ B$ sont les coups initiaux de $B$ et les coups initiaux de $\uparrow A^{\perp} \parr B$ sont les paires $(\star, b)$ où $b$ est un coup initial de $B$. Les coups non initiaux de $A \multimap\!\!\!\!\!\triangleright\ B$ et de $\uparrow A^{\perp} \parr B$ sont les coups de $A$ et les coups non initiaux de $B$. On définit la fonction $f$

de $M_{\uparrow A^\perp \parr B}$ dans $M_{A \multimap B}$ par $f(\star, b) = b$, $f(a) = a$ pour $a \in M_A$ et $f(b) = b$ pour $b$ non initial dans $B$. On montre aisément que l'image d'une partie de $\uparrow A^\perp \parr B$ est une partie de $A \multimap B$ et de même pour l'inverse de $f$ donc $f$ est un $a$-isomorphisme.
- C'est juste l'isomorphisme précédent.
- $\downarrow\top$ est l'unique jeu joueur à un coup et une partie non vide, c'est-à-dire 1.
- Immédiat d'après l'égalité précédente et $\sharp\top = \top$ (proposition 3.22). □

On a également toutes les propriété duales des propositions 3.12, 3.22 et 7.11.

7.12. Définition (Stratégie centrale)
Une stratégie $\sigma : A \multimap B$ est *centrale*, noté $\sigma : A \overset{\bullet}{\multimap} B$, si dans chaque partie non vide de $\sigma$, le premier coup $J$, qui suit un coup initial (donc dans $B$), existe et est dans $A$.

De même, $\sigma : A_1 \parr \ldots \parr A_n$ est centrale en $A_i$ si dans chaque partie de $\sigma$, le premier coup $J$, qui suit un coup initial, est dans $A_i$.

7.13. Définition (Décalages de stratégies)

($\uparrow$) Soit $\sigma : A \multimap B$ une stratégie, par l'isomorphisme de la proposition 7.11 et le lemme 3.14, on obtient la stratégie $\uparrow\sigma : \uparrow A^\perp \parr B$.

($\downarrow$) Soit $\sigma : A \parr B$ une stratégie, on définit la stratégie $\downarrow_A\sigma : \uparrow A^\perp \overset{\bullet}{\multimap} B$ par $\downarrow_A\sigma = \{b\star \mid \vdash_B b\} \cup \{b \star as \mid (a, b)s \in \sigma\}$.

(!) Soit $\sigma : A \parr B$ une stratégie, on définit la stratégie $!_A\sigma : ?A^\perp \overset{\bullet}{\multimap} B$ par $!_A\sigma = \{s \in \mathcal{P}^J_{?A^\perp \multimap B} \mid \forall b$ initial dans $B, \forall a$ initial dans $A, s \restriction_{b\star a} \in \downarrow_A\sigma\} : ?A^\perp \overset{\bullet}{\multimap} B$

7.14. Remarque : Les deux notions de centralité coïncident par l'isomorphisme de la proposition 7.11. En effet, si $\sigma$ est une stratégie sur $P^\perp \multimap A_1 \parr \ldots \parr A_n$, alors on a la stratégie $\uparrow\sigma : \uparrow P \parr A_1 \parr \ldots \parr A_n$ et $\sigma$ est centrale *ssi* $\uparrow\sigma$ est centrale en $\uparrow P$.

7.15. Lemme
*Si $s \in \mathcal{P}^O_{(A\parr C)\multimap(B\parr D)}$ et $s \notin L_{B\parr D}$ alors :*
- *soit $s \restriction_{A\multimap B} \in \mathcal{P}^O_{A\multimap B}$ et $s \restriction_{C\multimap D} \in \mathcal{P}^J_{C\multimap D}$,*
- *soit $s \restriction_{A\multimap B} \in \mathcal{P}^J_{A\multimap B}$ et $s \restriction_{C\multimap D} \in \mathcal{P}^O_{C\multimap D}$.*

Démonstration : On raisonne par récurrence sur $|s|$. Sachant que $|s|$ est impair et $|s| \neq 1$ par hypothèse, on a $s = tm^J n^O$ avec $t \in \mathcal{P}^O_{(A\parr C)\multimap(B\parr D)}$.
- Si $t \in L_{B\parr D}$, on a $m, n \in M_{A\parr C}$ (par *condition de switching* pour l'*arène flèche*) et $m = (a_1, c_1)$. Si $n \in M_A$ alors $s \restriction_{A\multimap B} = t \restriction_B a_1 n \in \mathcal{P}^O_{A\multimap B}$ et $s \restriction_{C\multimap D} = t \restriction_D c_1 \in \mathcal{P}^J_{C\multimap D}$, et le contraire si $n \in M_C$.
- Si $t \notin L_{B\parr D}$, supposons que l'hypothèse de récurrence nous donne $t \restriction_{A\multimap B} \in \mathcal{P}^O_{A\multimap B}$ et $t \restriction_{C\multimap D} \in \mathcal{P}^J_{C\multimap D}$ (l'autre cas se traitant de la même manière). On ne peut pas avoir $m^J \in M_C$ ni $m^J \in M_D$ (à cause de l'alternance des joueurs dans $tm \restriction_{C\multimap D}$).
  - Si $n^O \in A \multimap B$ alors $tmn \restriction_{A\multimap B} \in \mathcal{P}^O_{A\multimap B}$ et $tmn \restriction_{C\multimap D} = t \restriction_{C\multimap D} \in \mathcal{P}^J_{C\multimap D}$.
  - Si $n^O \in C \multimap D$ alors $tmn \restriction_{A\multimap B} = t \restriction_{A\multimap B} m \in \mathcal{P}^J_{A\multimap B}$ et $tmn \restriction_{C\multimap D} = t \restriction_{C\multimap D} n \in \mathcal{P}^O_{C\multimap D}$. □

7.16. Proposition ($\parr$ de stratégies et centralité)
*Soit $\sigma : A \overset{\bullet}{\multimap} B$ une stratégie centrale et $\tau : C \multimap D$ une stratégie quelconque, d'une part $\sigma \parr \tau = \{s \in \mathcal{P}^J_{(A\parr C)\multimap(B\parr D)} \mid s \restriction_{A\multimap B} \in \sigma \wedge s \restriction_{C\multimap D} \in \tau\}$ est une stratégie sur $(A \parr C) \multimap (B \parr D)$ et*

*d'autre part le diagramme suivant commute :*

$$\begin{array}{ccc} A \parr C & \xrightarrow{\sigma \parr C} & B \parr C \\ {\scriptstyle A\parr\tau}\downarrow & & \downarrow{\scriptstyle B\parr\tau} \\ A \parr D & \xrightarrow[\sigma \parr D]{} & B \parr D \end{array}$$

*où* $\sigma \parr C = \sigma \parr id_C$.

Démonstration : On montre que $\sigma \parr \tau$ est une stratégie :

- On a immédiatement que $\sigma \parr \tau$ est non vide et close par $J$-préfixe.
- Le point le plus délicat est le déterminisme. Soit $sm^J, sn^J \in \sigma \parr \tau$, on considère différents cas pour $s$ sachant que $s \neq \varepsilon$ puisqu'elle termine par un coup $O$ donc $s$ commence par un coup $(b_1, d_1)$. On considère deux cas :
  - Si $s \in L_{B\parr D}$ alors $s \restriction_{A \multimap B} = b_1^O$ (par centralité de $\sigma$) et $s \restriction_{C \multimap D} = d_1 \dots d_k^O$. Si $m \in M_{B\parr D}$ (resp. $n \in M_{B\parr D}$) alors $m = d_{k+1} \in M_D$ (resp. $n = d'_{k+1}$) et si $m \in M_{A\parr C}$ (resp. $n \in M_{A\parr C}$) alors $m = (a_1, c_1)$ (resp. $n = (a'_1, c'_1)$). On en déduit que $sm \restriction_{C \multimap D} = d_1 \dots d_k d_{k+1}^J$ ou $d_1 \dots d_k c_1^J$ et $sn \restriction_{C \multimap D} = d_1 \dots d_k {d'_{k+1}}^J$ ou $d_1 \dots d_k {c'_1}^J$, par déterminisme de $\tau$, ceci entraîne que $m = d_{k+1} = d'_{k+1} = n$ ou $c_1 = c'_1$ avec $m = (a_1, c_1)$ et $n = (a'_1, c_1)$, par déterminisme de $\sigma$, on conclut alors que $a_1 = a'_1$ et donc dans tous les cas $m = n$.
  - Si $s \notin L_{B\parr D}$ alors $|s| \geq 1$ donc $s = tm_1^O$. Si $m_1^O \in M_{A \multimap B}$ alors, par le lemme 7.15, $tm_1^O \restriction_{A \multimap B} \in \mathcal{P}^O_{A \multimap B}$ et $tm_1^O \restriction_{C \multimap D} \in \mathcal{P}^J_{C \multimap D}$ donc $m$ et $n$ sont dans $A \multimap B$ et, par déterminisme de $\sigma$, $m = n$ (de même si $m_1 \in M_{C \multimap D}$ par déterminisme de $\tau$).
- Il reste à prouver l'innocence de $\sigma \parr \tau$. Si $sm^O n^J \in \sigma \parr \tau$, $t \in \sigma \parr \tau$, $tm \in \mathcal{P}_{(A\parr C) \multimap (B \parr D)}$ et $\ulcorner sm \urcorner = \ulcorner tm \urcorner$ alors $smn \restriction_{A \multimap B}$, $t \restriction_{A \multimap B}$, $smn \restriction_{C \multimap D}$ et $t \restriction_{C \multimap D}$ vérifient les propriétés correspondantes donc, par innocence de $\sigma$ et $\tau$, on a $tmn \restriction_{A \multimap B} \in \sigma$ et $tmn \restriction_{C \multimap D} \in \tau$, d'où $tmn \in \sigma \parr \tau$.

Il est alors aisé de voir que $(\sigma \parr C);(B \parr \tau) = \sigma \parr \tau = (A \parr \tau);(\sigma \parr D)$. □

On a donné les constructions linéaires de jeux polarisés et on va désormais ajouter la structure nécessaire à l'interprétation des règles structurelles généralisées.

7.17. Définition (Jeu multiple)
Un jeu $A$ est *multiple* s'il est *bien ouvert* et si :

(m1) Pour toute partie $s \in \mathcal{P}_A$, de coup initial $m$ et pour tout coup $n$ de $s$ justifié par $m$, on a $s \restriction_{mn} \in \mathcal{P}_A$, où $s \restriction_{mn}$ est la sous-suite de $s$ commençant par $mn$ suivi des coups héréditairement justifiés par $n$.

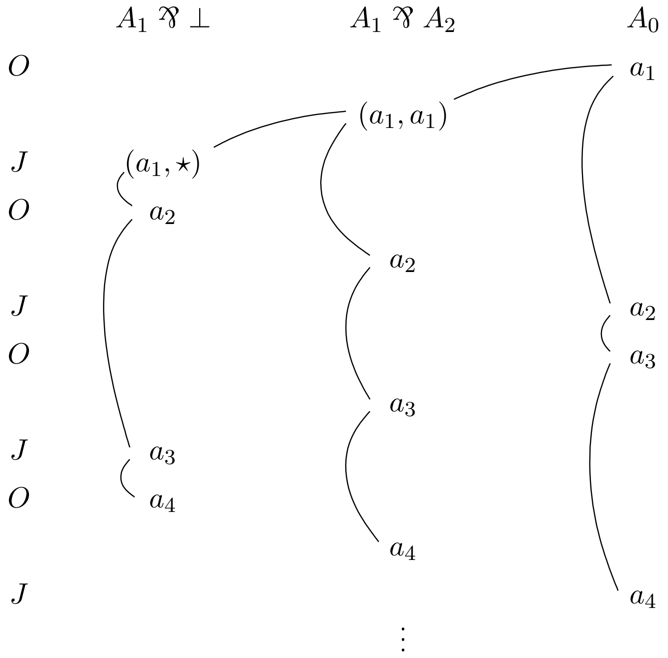
(m2) Pour toute position légale bien ouverte $s \in L_A$ de coup initial $m$, si $I + J$ est une partition des coups de $s$ justifiés par $m$ et si $s \restriction_{mI} \in \mathcal{P}_A$ et $s \restriction_{mJ} \in \mathcal{P}_A$, alors on a $s \in \mathcal{P}_A$.

7.18. Remarque : La multiplicité d'un jeu ne parle pas de sa polarité et $A$ est multiple *ssi* $A^\perp$ est multiple.

7.19. Lemme (Structure monoïdale des jeux multiples)
*Tout jeu négatif multiple a une structure de $\parr$-monoïde.*

Démonstration : Soit $A$ un jeu négatif multiple, si $s$ est une partie du jeu $A_1 \parr A_2 \multimap A_0$ (les indices ne servant qu'à distinguer les occurrences de $A$), on note $s_i$ ($i \in \{1, 2\}$) la sous-suite de $s$ contenant les coups de $A_i$ et les coups de $A_0$ tels que le dernier coup avant celui-ci qui ne soit pas dans $A_0$ est dans $A_i$. On définit alors la stratégie de contraction sur ce jeu par $c_A = \{s \in \mathcal{P}^J_{A_1 \parr A_2 \rightarrowtriangle A_0} \mid \forall t \leq^J s, t_1 \in id_A \wedge t_2 \in id_A\}$. La stratégie d'affaiblissement est définie sur le jeu $\perp \rightarrowtriangle A$ par $w_A = \{\varepsilon\} \cup \{a\star \mid a \in M^i_A\}$.

(NEUTRALITÉ) Soit $s \in \mathrm{int}(A_1 \parr \perp, A_1 \parr A_2, A_0)$ telle que $s \upharpoonright_{A_1 \parr \perp \rightarrowtriangle A_1 \parr A_2} \in A \parr w_A$ et $s \upharpoonright_{A_1 \parr A_2 \rightarrowtriangle A_0} \in c_A$, $s$ est de la forme :




donc $s \upharpoonright_{A_1 \parr \perp}$ est dans la stratégie sur $A \parr \perp \rightarrowtriangle A$ induite par l'isomorphisme $A \parr \perp \simeq A$. Et réciproquement, si $s$ est dans cette stratégie, on peut construire une suite d'interaction de la forme ci-dessus qui donne une partie dans $(A \parr w_A); c_A$.

(ASSOCIATIVITÉ) Tout comme on a défini la contraction sur $A \parr A \rightarrowtriangle A$, on peut définir une version ternaire sur $A \parr A \parr A \rightarrowtriangle A$ et on voit facilement qu'elle correspond à $(A \parr c_A); c_A$ et à $(c_A \parr A); c_A$ modulo associativité de $\parr$.

(COMMUTATIVITÉ) Immédiat. $\square$

7.20. REMARQUE : La stratégie de contraction $A \parr A \rightarrowtriangle A$ correspond à une stratégie sur $A \parr \uparrow(A^\perp \otimes A^\perp)$ et la stratégie d'affaiblissement $\perp \rightarrowtriangle A$ à une stratégie sur $A \parr \uparrow 1$. Ceci munit donc, par dualité, les jeux positifs multiples d'une structure de $\otimes$-comonoïde pour la notion appropriée de flèche : $P \to Q = P \multimap \uparrow Q$.

7.21. PROPOSITION (JEUX NÉGATIFS MULTIPLES)
*Soient $A$ et $B$ deux jeux négatifs,*
- *$\top$ et $\perp$ sont des jeux multiples.*
- *$?A^\perp$ est un jeu multiple.*
- *Si $A$ et $B$ sont multiples alors $A \parr B$ et $A \,\&\, B$ sont multiples.*

Démonstration :
- Le jeu $\top$ n'ayant pas de partie non vide, le résultat est immédiat.
- Les parties de $\perp$ ayant au plus un coup, (M1) et (M2) sont immédiatement vérifiées.

– Le seul coup initial de $?A^\perp$ est $\star$ donc une position légale non vide de ce jeu est de la forme $\star s$ avec $s \in L_A$. Soit $\star s \in \mathcal{P}_{?A^\perp}$, et soit $n$ un coup de $s$ justifié par $\star$, on a $(\star s) \upharpoonright_{\star n} = \star(s \upharpoonright_n)$ or, par définition de $\flat A^\perp$, $s \upharpoonright_n \in \mathcal{P}_{A^\perp} \subset \mathcal{P}_{\flat A^\perp}$ donc $(\star s) \upharpoonright_{\star n} \in \mathcal{P}_{?A^\perp}$, ce qui valide (M1). Soit $\star s \in L_{?A^\perp}$ et $I + J$ une partition des coups justifiés par $\star$ telle que $(\star s) \upharpoonright_{\star I} \in \mathcal{P}_{?A^\perp}$ et $(\star s) \upharpoonright_{\star J} \in \mathcal{P}_{?A^\perp}$, on a $s \upharpoonright_I \in \mathcal{P}_{\flat A^\perp}$ et $s \upharpoonright_J \in \mathcal{P}_{\flat A^\perp}$ donc $s \in \mathcal{P}_{\flat A^\perp}$ puisque $I + J$ est une partition des coups initiaux de $s$, d'où $\star s \in \mathcal{P}_{?A^\perp}$ ce qui valide (M2).
– Soit $s$ une partie de $A \parr B$ de coup initial $(a, b)$ et soit $c$ un coup justifié par $(a, b)$ dans $s$, supposons que $c \in M_A$, $s \upharpoonright_{(a,b)c}$ est obtenu à partir de $s \upharpoonright_{(a,b)c} \upharpoonright_A$ en remplaçant le premier coup $a$ par $(a, b)$ puisque tous les coups héréditairement justifiés par $c$ dans $s$ sont dans $A$. Or $A$ est multiple donc $s \upharpoonright_{(a,b)c} \upharpoonright_A = s \upharpoonright_A \upharpoonright_{ac}$ est une partie de $A$ d'où $s \upharpoonright_{(a,b)c} \in \mathcal{P}_{A \parr B}$. Soit $s$ une position légale de $A \parr B$ de coup initial $(a, b)$ et soit $I + J$ une partition des coups justifiés par $(a, b)$ telle que $s \upharpoonright_{(a,b)I} \in \mathcal{P}_{A \parr B}$ et $s \upharpoonright_{(a,b)J} \in \mathcal{P}_{A \parr B}$. On a $s \upharpoonright_{(a,b)I} \upharpoonright_A = s \upharpoonright_A \upharpoonright_{aI'}$ où $I' = I \cap M_A$ et de même avec $J$, or $I' + J'$ est une partition des coups justifiés par $a$ dans $s \upharpoonright_A$ donc par (M2) pour $A$, $s \upharpoonright_A \in \mathcal{P}_A$. En raisonnant de la même manière, on obtient $s \upharpoonright_B \in \mathcal{P}_B$ et donc $s \in \mathcal{P}_{A \parr B}$.
– Soit $s$ une partie de $A \& B$, on considère le cas où $s$ est une partie de $A$, et la propriété (M1) pour $A$ donne la propriété (M1) pour $A \& B$. Soit $s$ une position légale de $A \& B$ de coup initial $m$ et $I + J$ une partition des coups de $s$ justifiés par $m$ telle que $s \upharpoonright_{mI} \in \mathcal{P}_{A\&B}$ et $s \upharpoonright_{mJ} \in \mathcal{P}_{A\&B}$, supposons que $m \in M_A$, on a alors que $s \upharpoonright_{mI} \in \mathcal{P}_A$ et $s \upharpoonright_{mJ} \in \mathcal{P}_A$, et $s \in L_A$ puisque tout coup de $s$ est héréditairement justifié par $m$ donc est dans $A$. La propriété (M2) pour $A$ donne alors que $s \in \mathcal{P}_A \subset \mathcal{P}_{A\&B}$. □

7.22. Proposition (Décomposition additive)
*Soit $N$ une jeu négatif multiple, il existe des jeux négatifs bien ouverts $(N_i)_{i \in I}$ tels que $N \simeq_p \mathop{\&}\limits_{i \in I} ?N_i^\perp$ (avec la convention que l'on obtient $\top$ si $I = \emptyset$).*

**Démonstration :** On considère $I$ l'ensemble des coups initiaux de $N$ utilisés dans une partie de $N$ et pour tout $i \in I$ on note $N_i$ le jeu négatif défini par :
– $M_{N_i}$ est l'ensemble des coups $m$ de $N$ tels que $i$ valide héréditairement $m$.
– $\lambda_{N_i}(m) = \overline{\lambda_N}(m)$
– $\vdash_{N_i} m$ si $i \vdash_N m$
– $m \vdash_{N_i} n$ si $m \vdash_N n$
– $\mathcal{P}_{N_i} = \{s \in L_{N_i} \mid is \in \mathcal{P}_N \wedge s \text{ bien ouverte}\}$

On note $M = \mathop{\&}\limits_{i \in I} ?N_i^\perp$, pour montrer que $N \simeq_p M$, on montre que si on note $i$ le coup initial de $?N_i^\perp$, on a $\mathcal{P}_N = \mathcal{P}_M$.

Soient $is$ une partie non vide de $N$ ($i \in I$) et $J$ l'ensemble des coups de $s$ justifiés par $i$, pour tout $j \in J$, $is \upharpoonright_{ij} \in \mathcal{P}_N$ par multiplicité donc $s \upharpoonright_j \in \mathcal{P}_{N_i}$ d'où, par définition de $\flat$, $s \in \mathcal{P}_{?N_i^\perp}$.

Soit $s$ une partie non vide de $M$, il existe un $i \in I$ tel que $s = it \in \mathcal{P}_{?N_i^\perp}$. Par définition de ?, pour tout coup $j$ de $t$ justifié par $i$, $t \upharpoonright_j \in \mathcal{P}_{N_i}$ c'est-à-dire $it \upharpoonright_{ij} = i(t \upharpoonright_j) \in \mathcal{P}_N$. On en conclut, par multiplicité de $N$, que $s = it \in \mathcal{P}_N$. □

7.23. Définition (Construction $\hat{\parr}$)
On note $\psi_{A,B} : ?A^\perp \overset{\bullet}{\multimap} ?(A^\perp \otimes B^\perp) \parr B$ la stratégie définie par $\psi_{A,B} = !_A \uparrow d_{A \parr B}$.

Soit $\sigma : \sharp A \multimap B$ une stratégie, on définit $\sigma \hat{\parr} C : \sharp(A \parr C) \multimap (B \parr C)$ de la manière suivante : $\uparrow\sigma$ est une stratégie sur $?A^\perp \parr B$ que l'on compose avec $\psi_{A,C} \parr B$ pour obtenir une stratégie sur $?(A^\perp \otimes C^\perp) \parr C \parr B$ qui est, à $a$-isomorphisme près, une stratégie sur $\sharp(A \parr C) \multimap (B \parr C)$.

7.24. Théorème (Catégorie de contrôle de jeux)
*La catégorie dont les objets sont les jeux négatifs multiples et les morphismes de $A$ dans $B$ sont les stratégies sur $\sharp A \to B$ (ou de manière équivalente $!A \multimap B$) est une catégorie de contrôle.*

Démonstration : On a déjà donné les principaux éléments de ce résultat :

- Cette catégorie est cartésienne fermée par la proposition 3.25 puisque toutes les constructions préservent la multiplicité (proposition 7.21), en particulier l'exponentielle $!A \multimap B$ est multiple si $B$ l'est.
- Tout objet est muni d'une structure de monoïde, par le lemme 7.19 (en composant avec les dérélictions nécessaires).
- Cette catégorie est distributive, par la proposition 3.12 (en composant avec les dérélictions nécessaires).

Il reste à montrer les propriétés de $s_{A,B,C} : (?A^\perp \parr B) \parr C \to ?A^\perp \parr (B \parr C)$ et pour ce faire on va montrer que $s_{A,B,C}$ est tout simplement l'associativité de $\parr$ ce qui entraîne que c'est un isomorphisme et donne directement la commutation des diagrammes. Ceci s'obtient en curryfiant le diagramme commutatif suivant :

$$\begin{tikzcd}
 & (?A^\perp \parr B \parr C) \,\&\, A \arrow[dl, "d'_{A,?A^\perp \parr B,C}"'] \arrow[dr, "d'_{A,?A^\perp,B\parr C}"] & \\
((?A^\perp \parr B) \,\&\, A) \parr C \arrow[rr, "d'_{A,?A^\perp,B} \parr C"] \arrow[d, "ev \parr C"'] & & (?A^\perp \,\&\, A) \parr B \parr C \arrow[d, "ev \parr B \parr C"] \\
B \parr C & & \perp \parr (B \parr C) \arrow[ll, "l_{B\parr C}"]
\end{tikzcd}$$

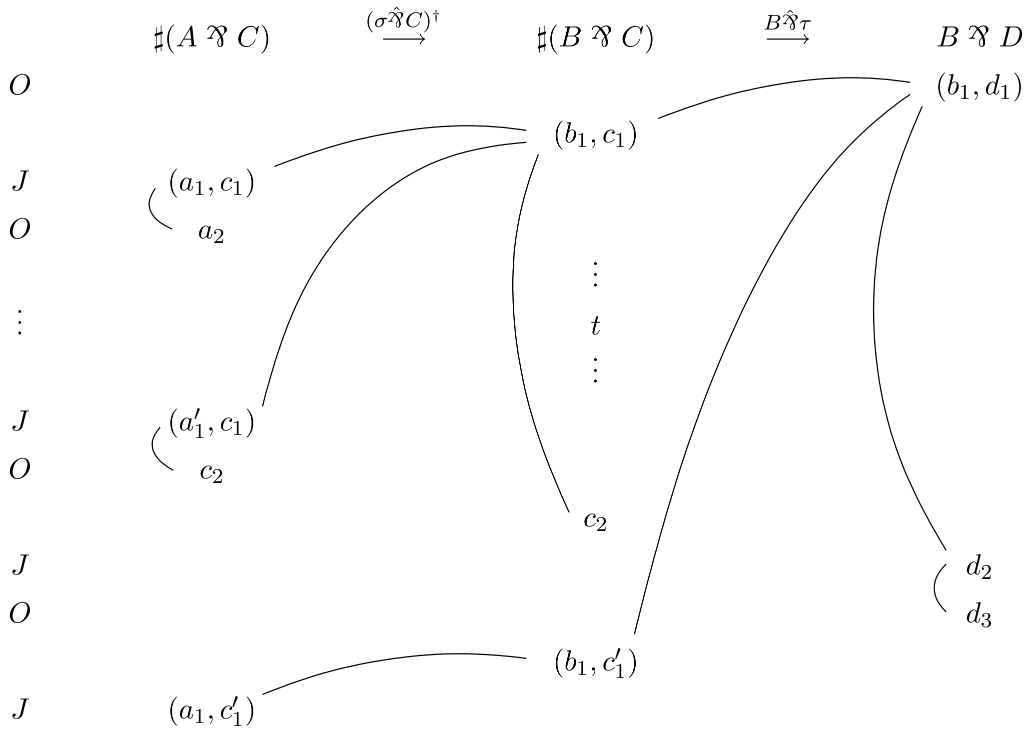
le triangle commute par propriétés de $d'_{A,B,C}$ et le carré par définition de $ev$. □

7.25. Proposition (Morphismes centraux)
*Soit $\sigma : A \to B$ un morphisme de la catégorie des jeux négatifs multiples, $\sigma$ est un morphisme central de la catégorie* ssi $\sigma = d_A; \sigma_0 : \sharp A \overset{\bullet}{\to} B$ *avec $\sigma_0$ stratégie centrale sur $A \overset{\bullet}{\to} B$.*

7.26. Remarque : Attention, une stratégie centrale n'est pas nécessairement un morphisme central, comme le prouve la propriété, il faut également une contrainte de linéarité.

Démonstration : Si $\sigma \neq d_A; \sigma_0 : \sharp A \overset{\bullet}{\to} B$ avec $\sigma_0$ stratégie centrale sur $A \overset{\bullet}{\to} B$, on considère deux cas :

- Soit $\sigma$ n'est pas une stratégie centrale sur $\sharp A \overset{\bullet}{\to} B$ parce qu'elle contient la partie $b_1^O b_2^J$, on considère la stratégie non centrale $\tau : \sharp\perp \to ?1$ contenant les parties $\varepsilon$ et $\star^O m^J$ où $m$ est l'unique coup de 1. On considère les deux stratégies $(\sigma \parr \perp)^\dagger; (B \parr \tau)$ et $(A \parr \tau)^\dagger; (\sigma \parr ?1)$, elles répondent différemment au coup $(b_1, \star)$ puisque la première répond $m$ et la seconde répond $b_2$, donc $\sigma$ n'est pas un morphisme central.
- Soit $\sigma$ est centrale sur $\sharp A \overset{\bullet}{\to} B$ mais il existe une partie $s$ de $\sigma$ qui contient deux coups initiaux dans $\sharp A$. Cette partie $s$ peut s'écrire $b_1^O a_1^J a_2^O t {a'_1}^J \ldots$ où $a_1$ et $a'_1$ pointent sur $b_1$ et $t$ ne contient aucun coup initial dans $A$. On considère une stratégie $\tau : \sharp C \to D$ contenant la partie $d_1^O c_1^J c_2^O d_2^J d_3^O {c'_1}^J \ldots$ où $c_1$ et $c'_1$ pointent sur $d_1$. On a les compositions représentées dans la figure 7.1, d'où on conclue que $\sigma$ n'est pas un morphisme central.




et

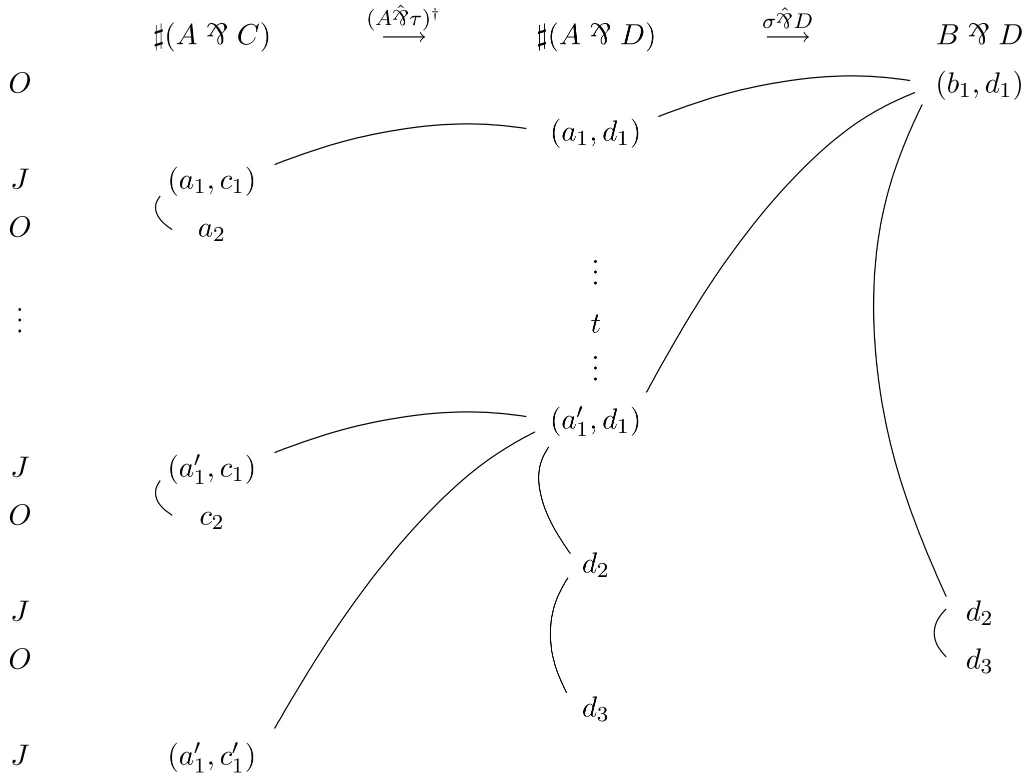
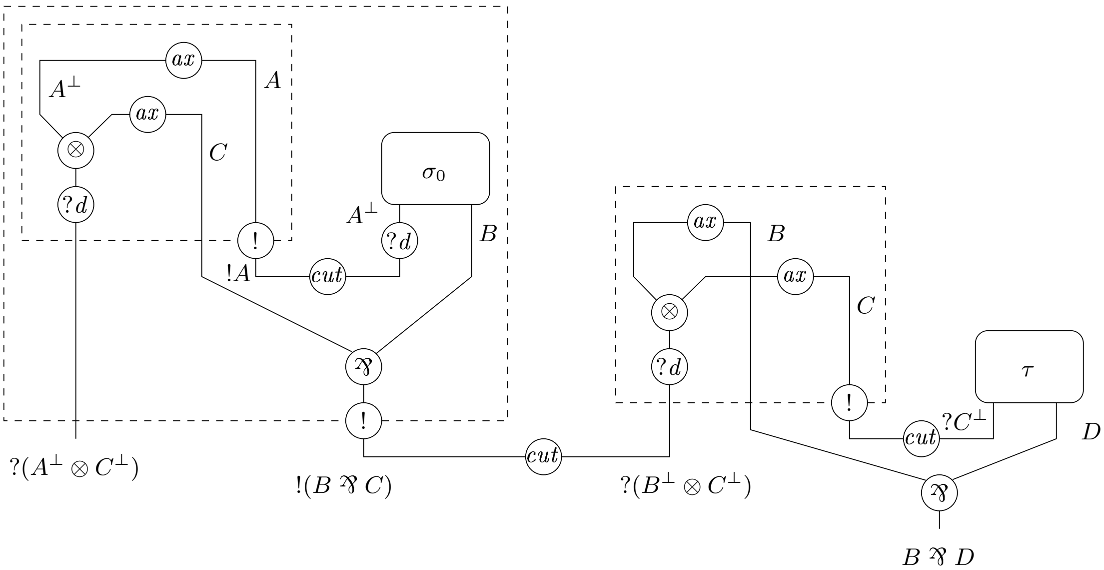



FIG. 7.1 – Stratégie centrale donnant un morphisme non central

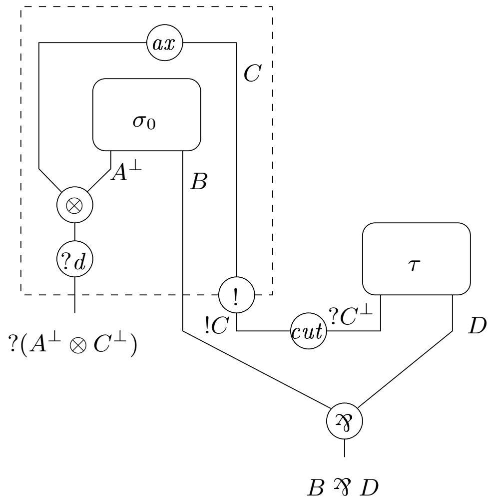
On montre maintenant la réciproque, si $\sigma = d_A; \sigma_0 : \sharp A \overset{\bullet}{\rightarrowtriangle} B$ avec $\sigma_0$ stratégie centrale sur $A \overset{\bullet}{\rightarrowtriangle} B$, soit $\tau : \sharp C \rightarrowtriangle D$ une stratégie quelconque, on doit prouver que $((d_A;\sigma_0)\hat{\parr}C)^\dagger; (B\hat{\parr}\tau) = (A\hat{\parr}\tau)^\dagger; ((d_A;\sigma_0)\hat{\parr}D)$ :

$$
\begin{aligned}
((d_A;\sigma_0)\hat{\parr}C)^\dagger; (B\hat{\parr}\tau) &= ((d_A\hat{\parr}C); (\sigma_0 \parr C))^\dagger; (B\hat{\parr}\tau)\\
(1a) &= (d_{A\parr C}; (\sigma_0 \parr C))^\dagger; (B\hat{\parr}\tau) : \sharp(A \parr C) \rightarrowtriangle B \parr D\\
&\simeq_a \uparrow(B\hat{\parr}\tau); !_{(B\parr C)}\uparrow(d_{A\parr C}; (\sigma_0 \parr C)) : ?(A^\perp \otimes C^\perp) \parr B \parr D\\
&= \uparrow\tau; (\psi_{C,B} \parr D); (!_{(B\parr C)}\uparrow(d_{A\parr C}; (\sigma_0 \parr C)) \parr D)\\
&= \uparrow\tau; (!_C\uparrow d_{B\parr C} \parr D); (!_{(B\parr C)}\uparrow(d_{A\parr C}; (\sigma_0 \parr C)) \parr D)\\
&= \uparrow\tau; (!_C\uparrow d_{B\parr C}; !_{(B\parr C)}\uparrow(d_{A\parr C}; (\sigma_0 \parr C)) \parr D)\\
(2a) &= \uparrow\tau; (!_C(\uparrow d_{B\parr C}; !_{(B\parr C)}\uparrow(d_{A\parr C}; (\sigma_0 \parr C))) \parr D)\\
(3a) &= \uparrow\tau; (!_C(\uparrow(d_{A\parr C}; (\sigma_0 \parr C))) \parr D)\\
&= \uparrow\tau; (!_C(\uparrow d_{A\parr C}; ?(A^\perp \otimes C^\perp) \parr \sigma_0 \parr C) \parr D)\\
(3b) &= \uparrow\tau; ((!_C\uparrow d_{A\parr C}; ?(A^\perp \otimes C^\perp) \parr \sigma_0) \parr D)\\
&= \uparrow\tau; ((\psi_{C,A}; ?(A^\perp \otimes C^\perp) \parr \sigma_0) \parr D)\\
&= \uparrow\tau; (\psi_{C,A} \parr D); (?(A^\perp \otimes C^\perp) \parr \sigma_0 \parr D)\\
&= \uparrow(A\hat{\parr}\tau); (?(A^\perp \otimes C^\perp) \parr \sigma_0 \parr D) : ?(A^\perp \otimes C^\perp) \parr B \parr D\\
&\simeq_a (A\hat{\parr}\tau); (\sigma_0 \parr D) : \sharp(A \parr C) \rightarrowtriangle B \parr D\\
(2b) &= (A\hat{\parr}\tau)^\dagger; d_{A\parr D}; (\sigma_0 \parr D)\\
(1b) &= (A\hat{\parr}\tau)^\dagger; (d_A\hat{\parr}D); (\sigma_0 \parr D)\\
&= (A\hat{\parr}\tau)^\dagger; ((d_A;\sigma_0)\hat{\parr}D)
\end{aligned}
$$

Ce calcul peut paraître, à juste titre, un peu rébarbatif, mais il s'avère beaucoup plus clair si on l'écrit à l'aide de réseaux. On considère $\sigma_0$ et $\tau$ comme des réseaux indéterminés (boîtes noires) de conclusions $A^\perp, B$ et $?C^\perp, D$. Le réseau associé à la stratégie $((d_A;\sigma_0)\hat{\parr}C)^\dagger; (B\hat{\parr}\tau)$ est alors :

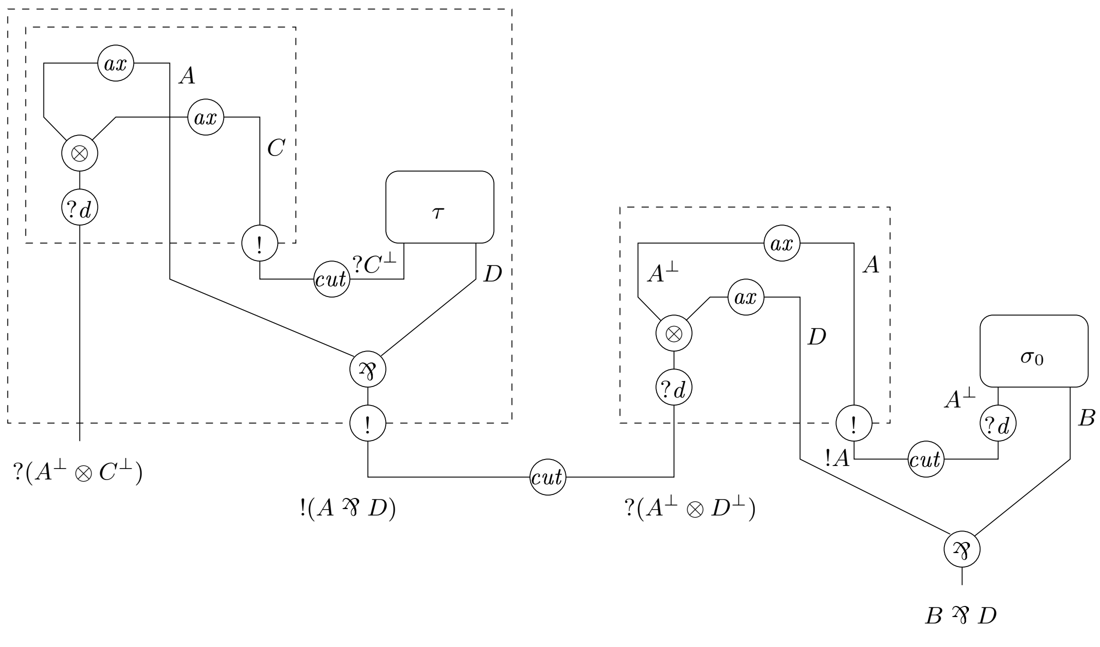


l'égalité ($1a$) correspond à la réduction $!/?d$ de la coupure de type $!A$, ($2a$) correspond à la réduction $!/!$ de la coupure de type $!(B \parr C)$ et ($3a$) correspond à la réduction $!/?d$ de la coupure de même type obtenue. On obtient ainsi le réseau :




qui correspond à la stratégie $\uparrow\tau; (!_C(\uparrow(d_{A\parr C}; (\sigma_0 \parr C))) \parr D)$.

En suivant désormais la preuve à partir du bas, le réseau associé à $(A\hat{\parr}\tau)^\dagger; ((d_A; \sigma_0)\hat{\parr}D)$ est :




l'égalité ($1b$) correspond à la réduction $!/?d$ de la coupure de type $!A$, ($2b$) correspond à la réduction $!/?d$ de la coupure de type $!(A \parr D)$ (l'arête de type $?A^\perp \otimes D^\perp$ n'étant plus porte auxiliaire de boîte) et ($3b$) correspond à la réduction $!/!$ de la coupure de type $A$ qui fait rentrer $\sigma_0$ dans la dernière boîte restante. On obtient ainsi également le réseau associé à $\uparrow\tau; (!_C(\uparrow(d_{A\parr C}; (\sigma_0 \parr C))) \parr D)$.

Les autres égalités sont des applications des définitions de $\uparrow$, $\parr$, $\hat{\parr}$ et $\psi_{A,B}$. $\square$

## 7.2 Modèle de jeux de $\mathsf{LLP}$

7.27. Définition (Stratégie gagnante)
Une stratégie $\sigma$ est *gagnante* si pour toute partie $s \in \sigma$ telle que $sm^O \in \mathcal{P}_A$ il existe un coup $n^J$ tel que $smn \in \sigma$.

### 7.2.1 Interprétation

On interprète les atomes négatifs $X^\perp$ par des jeux négatifs multiples quelconques et une fois ceci fixé, on interprète les formules par le jeu correspondant. Par la proposition 7.21, les formules négatives sont interprétées par des jeux négatifs multiples.

Si $\pi$ est une preuve du séquent $\vdash A_1, \ldots, A_n$ ne contenant pas de formule positive, $\pi$ est interprétée par une stratégie gagnante sur le jeu $A_1 \parr \ldots \parr A_n$. Si $\pi$ est une preuve du séquent $\vdash P, A_1, \ldots, A_n$, $\pi$ est interprétée par une stratégie gagnante sur le jeu $\uparrow P \parr A_1 \parr \ldots \parr A_n$ centrale en $\uparrow P$ (si $n = 0$, on obtient une stratégie sur $\uparrow P$).

On regarde chaque cas de dernière règle de $\pi$ :

($ax$) La stratégie $\uparrow id_A : \uparrow A^\perp \parr A$ centrale en $\uparrow A^\perp$ interprète la preuve.

($cut$) Soit $\sigma : A \parr B$ la stratégie interprétant la première prémisse et $\tau : B \overset{\bullet}{\rightarrow} C$ la stratégie obtenue par isomorphisme à partir de l'interprétation de la seconde prémisse, la preuve est interprétée par $\sigma; A \parr \tau$.

($\otimes$) À isomorphisme près les prémisses donnent deux stratégies $\sigma : A \overset{\bullet}{\rightarrow} C$ et $\tau : B \overset{\bullet}{\rightarrow} D$, on interprète la preuve par $\uparrow(\sigma \parr \tau) : \uparrow(A^\perp \otimes B^\perp) \parr C \parr D$.

($\parr$) La prémisse de la règle et la preuve tout entière sont interprétées de la même manière.

($\oplus_i$) La prémisse donne une stratégie $\sigma : \uparrow A_i^\perp \parr B$, on interprète la preuve par $\sigma : \uparrow(A_1^\perp \oplus A_2^\perp) \parr B$.

($\&$) Les prémisses donne deux stratégies $\sigma : A \parr C$ et $\tau : B \parr C$, on interprète la preuve par $<\sigma, \tau>; {d_{C,A,B}}^{-1} : (A \,\&\, B) \parr C$.

(1) La preuve est interprétée par l'unique stratégie non vide sur le jeu 1.

($\perp$) Par isomorphisme, la stratégie obtenue sur $A$ donne une stratégie sur $A \parr \perp$.

($\top$) La preuve est interprétée par la stratégie vide sur le jeu vide $A \parr \top$.

(!) Soit $\sigma : A \parr B$ la stratégie obtenue à partir de la prémisse, la preuve est interprétée par $\uparrow !_A \sigma : \uparrow !A \parr B$.

($?d$) La prémisse donne une stratégie $\sigma : \uparrow A^\perp \parr B$, on interprète la preuve par $\sigma : \uparrow \flat A^\perp \parr B$.

($?c$) Puisque tout jeu négatif multiple a une structure de $\parr$-monoïde, si $\sigma : B \parr A \parr A$ est la stratégie obtenue pour la prémisse, on interprète la preuve par $\sigma; (B \parr c_A) : B \parr A$.

($?w$) Puisque tout jeu négatif multiple a une structure de $\parr$-monoïde, si $\sigma : B$ est la stratégie obtenue pour la prémisse et $\sigma'$ la stratégie sur $B \parr \perp$ obtenue par isomorphisme à partir de $\sigma$, on interprète la preuve par $\sigma'; (B \parr w_A) : B \parr A$.

7.28. Proposition (Correction)
*Par l'interprétation précédente, les jeux polarisés fournissent un modèle dénotationnel de* $\mathsf{LLP}$.

On peut voir ce résultat comme un corollaire des théorèmes 6.21 et 7.24 ou le redémontrer directement (voir[Lau02a]).

### 7.2.2 Définissabilité

On va montrer une réciproque de la proposition 7.28. Ces résultats sont connus sous le nom de *complétude forte* ou *définissabilité*.

7.29. Lemme (Décomposition additive (bis))
*Soit $A$ le jeu interprétant une formule négative sans atome, il existe des formules négatives $(A_i)_{1\leq i\leq n}$ sans atome telles que $A \simeq_p \mathop{\&}\limits_{1\leq i\leq n} ?A_i^\perp$ (avec la convention que l'on obtient $\top$ si $n=0$).*

Démonstration : Par récurrence sur la taille de $A$ :

- $A = ?B^\perp$, immédiat.
- si $A = \perp$ alors $A \simeq_p ?\top^\perp$.
- si $A = \top$ alors $n = 0$.
- $A = B \,\&\, C$, immédiat par hypothèse de récurrence sur $B$ et $C$.
- $A = B \parr C$, par hypothèse de récurrence $B \simeq_p \mathop{\&}\limits_{1\leq i\leq n} ?B_i^\perp$ et $C \simeq_p \mathop{\&}\limits_{1\leq j\leq m} ?C_j^\perp$ donc par $p$-distributivité de $\parr$ par rapport $\&$, on a

$$A \simeq_p \mathop{\&}\limits_{\substack{1\leq i\leq n\\1\leq j\leq m}} (?B_i^\perp \parr ?C_j^\perp)$$

De plus $?B_i^\perp \parr ?C_j^\perp \simeq_p ?(B_i \,\&\, C_j)^\perp$ d'où

$$A \simeq_p \mathop{\&}\limits_{\substack{1\leq i\leq n\\1\leq j\leq m}} ?(B_i \,\&\, C_j)^\perp$$

□

7.30. Remarque : Ce résultat a déjà été prouvé de manière un peu différente dans la proposition 4.28 puisque les isomorphismes syntaxiques utilisés sont des $p$-isomorphismes entre jeux. La proposition 7.22 ne suffit pas ici puisqu'elle permet de décomposer tout jeu négatif multiple mais ne peut pas être appliquée récursivement puisque rien ne garantit que les jeux apparaissant dans la décomposition soient eux-mêmes multiples.

Une stratégie déterministe, innocente et gagnante peut être vue comme une fonction des parties finissant par un coup $O$ et qui sont des vues joueurs (dont les coups de l'Opposant pointent toujours sur le coup qui précède) dans les coups $J$. Son graphe est ainsi formé de paires $(s, m^J)$ et c'est ce que l'on utilise dans la définition suivante :

7.31. Définition (Taille d'une stratégie)
La *taille d'une stratégie* $|\sigma|$ est la somme des longueurs des vues joueurs de son graphe. Si $|\sigma|$ est fini, on dit que $\sigma$ est une *stratégie finie*.

7.32. Exemple (Stratégie infinie)
La stratégie formée des préfixes joueurs de la partie suivante sur le jeu $?!\perp$ (dont les trois coups seront notés $?$, $!$ et $\perp$) n'est pas finie :

$$\begin{array}{lc}
O & ? \\
J & ! \\
O & \bot \\
J & ! \\
O & \bot \\
J & ! \\
O & \bot \\
 & \vdots
\end{array}$$

elle est pourtant déterministe, innocente et gagnante.

Cette stratégie n'est pas l'interprétation d'une preuve de LLP, elle serait l'interprétation de la pseudo-preuve infinie suivante :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\\ \vdash ?!\bot\end{array}}{\vdash \bot, ?!\bot}\,\bot}{\vdash !\bot, ?!\bot}\,!}{\vdash ?!\bot, ?!\bot}\,?d}{\vdash ?!\bot}\,?c}{\vdash \bot, ?!\bot}\,\bot}{\vdash !\bot, ?!\bot}\,!}{\vdash ?!\bot, ?!\bot}\,?d}{\vdash ?!\bot}\,?c$$

7.33. Lemme ($\downarrow\uparrow$-lemme)
*Soit* $\sigma : \uparrow A^\perp \overset{\bullet}{\multimap} \uparrow B^\perp$ *une stratégie, soit* $\sigma_0 = \{s \mid \star\star s \in \sigma\} : B \multimap A$*,* $\sigma_0$ *est une stratégie et* $\sigma = \downarrow_A \uparrow \sigma_0$*.*

Démonstration : Il suffit de remarquer que, par centralité, une partie de $\sigma$ commence nécessairement par le coup $\star$ de $\uparrow B^\perp$ suivi du coup $\star$ de $\uparrow A^\perp$. □

7.34. Lemme (!-lemme)
*Soit* $\sigma : ?A^\perp \overset{\bullet}{\multimap} ?B^\perp$ *une stratégie, on a* $\sigma = !_A(\uparrow d_A; \sigma \parr A)$ *et* $|\uparrow d_A; \sigma \parr A| < |\sigma|$*.*

Démonstration : Puisque $?A^\perp = \uparrow\flat A^\perp$, par le lemme 7.33, $\sigma = \downarrow_A\uparrow\sigma_0$ avec $\sigma_0 : \sharp B \multimap \sharp A$. Par le lemme 3.26, $\sigma_0 = (\sigma_0; d_A)^\dagger$ avec $\sigma_0; d_A : \sharp B \multimap A$. Il est facile de vérifier que $\uparrow(\sigma_0; d_A) = \uparrow d_A; (\sigma \parr A) : ?B^\perp \parr A$ et $\downarrow_A\uparrow\tau^\dagger = !_A\uparrow\tau$, d'où $\sigma = !_A(\uparrow d_A; (\sigma \parr A))$.

Pour ce qui est de la taille de $\sigma$, il suffit de remarquer que pour toute stratégie $\tau$, $|\tau| < |!\tau|$ puisque $|\tau| < |\downarrow\tau|$. □

7.35. Remarque : Ces deux derniers lemmes nous donnent la réversibilité de $\downarrow$ et celle de ! (obtenue comme composition de celles de $\downarrow$ et $\sharp$).

7.36. Lemme (Focalisation $\oplus$)
*Soit* $\sigma : (\underset{1\leq i\leq n}{\&} ?A_i^\perp) \overset{\bullet}{\multimap} ?B^\perp$ *une stratégie, il existe* $1 \leq i \leq n$ *tel que* $\sigma$ *est une stratégie sur* $?A_i^\perp \overset{\bullet}{\multimap} ?B^\perp$*.*

Démonstration : Le jeu $?B^\perp$ a un unique coup initial donc, par déterminisme de $\sigma$, le premier coup $J$ est toujours dans la même composante $i$ de $\underset{1\leq i\leq n}{\&} ?A_i^\perp$, d'où on déduit que $\sigma$ est également une stratégie sur $?A_i^\perp \overset{\bullet}{\multimap} ?B^\perp$. □

7.37. Remarque : L'interprétation d'une preuve de LLP de $\vdash \mathcal{N}, P$ est une stratégie centrale gagnante finie sur $P^\perp \overset{\bullet}{\multimap} (\parr \mathcal{N})$ et l'interprétation d'une preuve de $\vdash \mathcal{N}$ est une stratégie gagnante finie sur $\parr \mathcal{N}$. Le résultat suivant est donc le plus précis que l'on puisse obtenir, séparant les cas des stratégies centrales et non centrales.

7.38. Théorème (Définissabilité propositionnelle)
*Soient $A$ et $B$ des formules négatives sans atome :*
- *Si $\sigma$ est une stratégie centrale gagnante finie sur $B \overset{\bullet}{\multimap} A$ alors $\sigma$ est l'interprétation d'une preuve de $\vdash B^\perp, A$ dans* LLP*.*
- *Si $\sigma$ est une stratégie gagnante finie sur $A$ alors $\sigma$ est l'interprétation d'une preuve de $\vdash A$ dans* LLP*.*

Démonstration : Grâce au lemme 7.29, on peut se restreindre aux formules de la forme $\underset{1\leq i\leq n}{\&} ?A_i^\perp$ et $(\underset{1\leq j\leq m}{\&} ?B_j^\perp) \multimap (\underset{1\leq i\leq n}{\&} ?A_i^\perp)$.

On raisonne par récurrence sur le couple $(|\sigma|, |B| + |A|)$. On se ramène tout d'abord au cas où $n = 1$ :
- Si $n = 0$, le jeu est vide dans les deux cas et $\sigma$ également, c'est l'interprétation d'une règle $\top$.
- Si $n > 1$, on note $\sigma_i = \sigma \upharpoonright_{?A_i^\perp}$ (resp. $\sigma \upharpoonright_{(\underset{1\leq j\leq m}{\&} ?B_j^\perp) \multimap ?A_i^\perp}$) qui est une stratégie définissable par hypothèse de récurrence ($|\sigma_i| < |\sigma|$) et puisque $\sigma = \bigcup_{1\leq i\leq n} \sigma_i$, elle est définissable par des règles &.

On montre que l'on peut également se ramener à $m = 1$ si $\sigma : (\underset{1\leq j\leq m}{\&} ?B_j^\perp) \overset{\bullet}{\multimap} ?A^\perp$ :
- Si $m = 0$, $\sigma$ ne peut pas être centrale.
- Si $m > 1$, par le lemme 7.36, il existe $j$ tel que $\sigma$ est une stratégie sur $?B_j^\perp \overset{\bullet}{\multimap} A$ et est définissable par hypothèse de récurrence puisque $|?B_j^\perp| < |\underset{1\leq j\leq m}{\&} ?B_j^\perp|$. On en conclut que $\sigma$ est définissable en tant que stratégie sur $B \overset{\bullet}{\multimap} A$ grâce à des règles $\oplus$.

On est donc ramené au cas où $\sigma$ est une stratégie sur $?A^\perp$ ou sur $?B^\perp \overset{\bullet}{\multimap} ?A^\perp$. Pour le second cas, par le lemme 7.34, il suffit de prouver la définissabilité de $\uparrow d_B; \sigma \parr B : ?A^\perp \parr B$ ce qui s'obtient par hypothèse de récurrence.

Reste donc le cas $\sigma : ?A^\perp$. Si chaque partie de $\sigma$ ne contient qu'un coup justifié par le coup initial, $\sigma$ est, à isomorphisme près, une stratégie sur $A \overset{\bullet}{\multimap} \perp$ définissable par hypothèse de récurrence ($|A| + 1 < |\uparrow\flat A^\perp|$). Sinon, il existe une partie de $\sigma$ ayant deux coups justifiés par le coup initial, on définit la stratégie $\sigma_1 : \uparrow A_1^\perp \parr ?A_2^\perp$ (où les indices servent juste à séparer les occurrences de $A$) par : si $s$ est une partie de $\sigma$, on obtient une partie de $\sigma_1$ en mettant dans $A_1$ le premier coup joueur et tous les coups justifiés par lui et dans $A_2$ les autres. On a clairement $\sigma = \sigma_1; c_{?A^\perp}$ et de plus $\sigma_1 = \uparrow\sigma_2$ où $\sigma_2 : A_1 \overset{\bullet}{\multimap} ?A_2^\perp$. Par les lemmes 7.29 et 7.36, on obtient une stratégie $\sigma_3 : ?A_j^\perp \overset{\bullet}{\multimap} ?A^\perp$. Enfin en appliquant le lemme 7.34 à $\sigma_3$, on obtient une stratégie $\sigma_4 : ?A^\perp \parr A_j$ plus petite que $\sigma$ (ce qui n'est pas nécessairement le cas pour $\sigma_1$, $\sigma_2$ et $\sigma_3$ lorsque $A = ?B^\perp$) et donc définissable par hypothèse de récurrence. □

7.39. Remarque : Dans ce cadre sans atome, la définissabilité dans LLP équivaut à la définissabilité dans $\mathsf{LL}_{\mathrm{pol}}$ comme le montre la preuve du théorème.

7.40. Exemple (Stratégie `catch`)
Pour illustrer ce résultat et les possibilités d'interprétation de primitives de contrôle des langages de programmation dans les jeux polarisés, on peut considérer la stratégie `catch` de type $(U \to U) \to U$ où $U$ est un jeu négatif à deux coups $q^O$ et $\checkmark^J$. Cette stratégie contient les deux parties suivantes et leurs $J$-préfixes :

$$
\begin{array}{ccccccccc}
 & ( & U & \to & U & ) & \to & U \\
O & & & & & & & q \\
J & & & & q & & & \\
O & & q & & & & & \\
J & & & & & & & \checkmark
\end{array}
$$

et

$$
\begin{array}{ccccccccc}
O & & & & & & & q \\
J & & & & q & & & \\
O & & & & \checkmark & & & \\
J & & & & & & & \checkmark
\end{array}
$$

$U$ est le jeu correspondant à la formule polarisée $?1$ donc, par définissabilité, `catch` est l'interprétation d'une preuve de $\mathsf{LL}_{\mathrm{pol}}$ de $!(!?1 \multimap ?1) \multimap ?1$ c'est-à-dire $?(!?1 \otimes !\bot) \parr ?1$. La stratégie étant centrale, on en déduit même que c'est l'interprétation d'une preuve de $\vdash !?1_1 \otimes !\bot, ?1_2$ (où les indices distinguent les occurrences). Cette preuve est :

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{$1$}
\UnaryInfC{$\vdash 1_2$}
\RightLabel{$?d$}
\UnaryInfC{$\vdash ?1_2$}
\RightLabel{$?w$}
\UnaryInfC{$\vdash ?1_1, ?1_2$}
\RightLabel{$!$}
\UnaryInfC{$\vdash !?1_1, ?1_2$}
\AxiomC{}
\RightLabel{$1$}
\UnaryInfC{$\vdash 1_2$}
\RightLabel{$?d$}
\UnaryInfC{$\vdash ?1_2$}
\RightLabel{$\bot$}
\UnaryInfC{$\vdash \bot, ?1_2$}
\RightLabel{$!$}
\UnaryInfC{$\vdash !\bot, ?1_2$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash !?1_1 \otimes !\bot, ?1_2, ?1_2$}
\RightLabel{$?c$}
\UnaryInfC{$\vdash !?1_1 \otimes !\bot, ?1_2$}
\end{prooftree}
$$

## 7.3 Jeux AJM polarisés

7.41. Définition (Jeu AJM polarisé)
Un *jeu AJM polarisé* $A$ est un quintuplet $(\pi_A, M_A, \lambda_A, \mathcal{P}_A, \equiv_A)$ où $(M_A, \lambda_A)$ est une arène AJM et $\mathcal{P}_A$, appelé ensemble des *parties* de $A$, est un ensemble de suites alternées de coups de $A$ commençant par un coup $m$ tel que $\lambda_A(m) = \pi_A$, enfin $\equiv_A$ est une relation d'équivalence entre parties de même longueur telle que :
- $s \equiv_A t \wedge s' \leq s \wedge t' \leq t \wedge |s'| = |t'| \Rightarrow s' \equiv_A t'$
- $s \equiv_A t \wedge sm \in \mathcal{P}_A \Rightarrow \exists n (tn \in \mathcal{P}_A \wedge sm \equiv_A tn)$

7.42. Remarque : Un jeu AJM polarisé négatif est exactement un jeu AJM au sens de la définition 3.29.

7.43. Définition (Constructions polarisées de jeux)
De manière similaire au cas HO, on définit les constructions suivantes :

($\uparrow$) Si $P$ est un jeu positif, le jeu négatif $\uparrow P$ est construit comme dans le cas HO et la relation d'équivalence entre parties est $\star s \equiv_{\uparrow P} \star t$ si $s \equiv_P t$.

($\perp$) On note $\perp$ le jeu ayant un coup $\star$ et les deux parties possibles : $(\{\star\}, \lambda_\perp(\star) = O, \{\varepsilon, \star\}, =)$.

($\parr$) Si $N$ et $M$ sont deux jeux négatifs, le jeu $N \parr M$ est construit comme dans le cas HO, excepté pour l'ensemble de parties qui est donné par $\mathcal{P}_{N \parr M} = \{\varepsilon\} \cup \{(n,m)s \mid n(s \upharpoonright_N) \in \mathcal{P}_N \wedge m(s \upharpoonright_M) \in \mathcal{P}_M\}$. La relation d'équivalence entre parties est $s \equiv_{N \parr M} t$ si $s \upharpoonright_N \equiv_N t \upharpoonright_N$, $s \upharpoonright_M \equiv_M t \upharpoonright_M$ et l'entrelacement des parties est le même. Contrairement au cas HO, la bonne ouverture au sens des jeux AJM (définition 3.37) ne pose ici aucun problème.

7.44. Remarque : Si $A$ et $B$ sont des jeux pour lesquels la relation d'équivalence entre parties est l'égalité, alors c'est également le cas pour $A \odot B$, $A \multimap B$, $A \& B$, $\downarrow A$ et $A \parr B$ (de même pour les constructions duales). L'introduction de cette relation d'équivalence n'est donc nécessaire que si l'on s'intéresse aux connecteurs exponentiels.

Afin d'obtenir un modèle de jeux polarisés pour LLP basé sur les jeux AJM, il nous faut introduire la notion adéquate de jeux multiples.

7.45. Définition (Jeux AJM multiples)
Un jeu AJM $A$ est *multiple* si ses coups non initiaux sont des paires commençant par un entier (appelé l'*indice*) et :

- Si $s \in \mathcal{P}_A$ et $i \in \mathbb{N}$, $s \upharpoonright_i \in \mathcal{P}_A$ (où $s \upharpoonright_i$ est la sous-suite de $s$ formée du premier coup et des coups d'indice $i$).
- Si $\varphi$ est une fonction injective de $\mathbb{N}$ dans $\mathbb{N}$ et $s \in \mathcal{P}_A$, alors $\varphi(s) \in \mathcal{P}_A$ et $s \equiv_A \varphi(s)$ (où $\varphi(s)$ est obtenue en appliquant $\varphi$ aux indices ce qui ne touche pas au premier coup).
- Si $s \in \mathcal{P}_A$ et $t \in \mathcal{P}_A$ ont le même premier coup et des ensembles d'indices disjoints alors tout entrelacement de $s$ et $t$ identifiant leur premier coup commun est une partie de $A$. De plus si $s \equiv_A s'$ et $t \equiv_A t'$ (avec $s'$ et $t'$ à indices disjoints) alors si $u$ est obtenue à partir de $s$ et $t$ et $u'$ à partir de $s'$ et $t'$ en les entrelaçant par le même entrelacement, on a $u \equiv_A u'$.

7.46. Lemme (Monoïdes)
*Tout jeu AJM négatif multiple a une structure de $\parr$-monoïde, et les stratégies de contraction et d'affaiblissement sont sans mémoire.*

Démonstration : Soit $A$ un jeu négatif multiple et soient $l$ et $r$ deux injections de $\mathbb{N}$ dans $\mathbb{N}$ à images disjointes, on définit la stratégie sans mémoire $c_A^{l,r}$ sur le jeu $A_1 \parr A_2 \overset{\bullet}{\multimap} A_0$ (où les indices ne servent qu'à distinguer les occurrences) par : (on note entre crochets l'occurrence dans laquelle le coup est joué)

$$
\begin{aligned}
a[A_0] &\mapsto (a,a)[A_1,A_2] \quad \text{si } a \text{ est initial} \\
(l(i),a)[A_0] &\mapsto (i,a)[A_1] \\
(r(i),a)[A_0] &\mapsto (i,a)[A_2] \\
(i,a)[A_1] &\mapsto (l(i),a)[A_0] \\
(i,a)[A_2] &\mapsto (r(i),a)[A_0]
\end{aligned}
$$

Il faut montrer que l'on obtient une stratégie. Soit $s$ une partie joueur induite par cette fonction, on a $l(s \upharpoonright_{A_1}) = s \upharpoonright_{A_0} \upharpoonright_{l(\mathbb{N})}$ et $r(s \upharpoonright_{A_2}) = s \upharpoonright_{A_0} \upharpoonright_{r(\mathbb{N})}$, par propriétés des jeux multiples, on en déduit que $s$ est une partie du jeu $A_1 \parr A_2 \multimap A_0$. Puisque $l$ et $r$ sont des injections et sont à images disjointes, on obtient que c'est bien une stratégie déterministe et sans mémoire.

En utilisant la définition des jeux multiples et comme pour la définition 3.35, on montre facilement que, si $(l', r')$ est une paire d'injections de $\mathbb{N}$ dans $\mathbb{N}$ à images disjointes, $c_A^{l,r} \approx c_A^{l',r'}$. Soit $w_A = \{\varepsilon\} \cup \{a\star \mid a \text{ initial dans } A\}$ la stratégie d'affaiblissement sur le jeu $\perp \overset{\bullet}{\multimap} A$, on montre comme pour les jeux HO que $A$ muni de la classe d'équivalence de $c_A^{l,r}$ et de celle de $w_A$ a une structure de $\parr$-monoïde. □

7.47. Proposition (Correction)
*En interprétant les formules négatives par des jeux AJM négatifs multiples et les preuves par des classes d'équivalence de stratégies* pas nécessairement sans mémoire*, les jeux polarisés AJM fournissent un modèle dénotationnel de* LLP.

Démonstration : C'est essentiellement la même preuve que pour les jeux HO en omettant les pointeurs et en utilisant les stratégies $c_A^{l,r}$ et $w_A$ pour les règles structurelles sur les jeux négatifs multiples. □

Il n'est pas possible de prouver de résultat de définissabilité pour les jeux AJM sans utiliser la contrainte que les stratégies soient *sans mémoire*. Malheureusement... les stratégies sans mémoire ne fournissent pas un modèle de LLP à cause des connecteurs additifs. On va désormais se restreindre à MELLP pour obtenir des résultats plus forts.

7.48. Définition (Jeu très bien ouvert)
Un jeu AJM $A$ est *très bien ouvert* s'il est bien ouvert et si toutes ses parties commencent par le même premier coup.

7.49. Lemme
*Le jeu AJM interprétant une formule de* MELLP *est un jeu très bien ouvert si les atomes sont interprétés par des jeux très bien ouverts.*

Démonstration : Immédiat pour les connecteurs 1 et !, le premier coup d'une partie de $A \otimes B$ où $A$ et $B$ sont très bien ouverts ne peut être que la paire des premiers coups des parties de $A$ et des parties de $B$. □

On considère désormais que les atomes sont interprétés par des jeux multiples très bien ouverts.

7.50. Lemme
*La stratégie AJM interprétant une preuve de* MELLP *est une stratégie sans mémoire.*

Démonstration : La seule construction qui ne préserve pas cette propriété de manière évidente est $!_A\sigma$. Soit $\sigma : A \parr B$ une stratégie sans mémoire, les parties de $!_A\sigma : ?A^\perp \overset{\bullet}{\multimap} B$ commencent par $b\star(i,a)(j,c)$ où $(j,c)$ est obtenu à partir de $i$ et de la réponse $(i',c)$ de $\sigma$ au coup $(a,b)$ or $a$ et $b$ sont les uniques coups initiaux de $A$ et $B$ donc le deuxième coup $(i',c)$ d'une partie de $\sigma$ est toujours le même (et ne dépend pas de $b$). Les deux premiers coups des parties de $!_A\sigma$ sont toujours les mêmes et ainsi $c$ est uniquement déterminé par la donnée de $(i,a)$ conformément à l'absence de mémoire. La suite des parties respecte la propriété d'être sans mémoire de manière immédiate puisque $\sigma$ est sans mémoire. □

7.51. Proposition (Correction sans mémoire pour MELLP)
*En interprétant les formules négatives de* MELLP *par des jeux AJM négatifs multiples très bien ouverts et les preuves par des classes d'équivalence de stratégies sans mémoire, les jeux polarisés AJM fournissent un modèle dénotationnel de* MELLP*.*

Démonstration : Application immédiate des deux lemmes précédents et de la proposition 7.47. □

7.52. Lemme (Réversibilité de $\otimes$)
*Soient $C$ l'interprétation d'une formule de* MELLP *sans atome et $\sigma : A \parr B \overset{\bullet}{\multimap} C$ une stratégie centrale sans mémoire, on a $\sigma \approx ((p_1;\sigma) \parr (p_2;\sigma)); c_C^{l,r}$ où $p_1 : A \overset{\bullet}{\multimap} A \parr B$ et $p_2 : B \overset{\bullet}{\multimap} A \parr B$ sont obtenues à partir de l'identité par affaiblissement sur $B$ et $A$ respectivement.*

Démonstration : Soit $s$ une partie de $A \parr B \multimap C$, si $c$ est un coup de $s$ dans $C$ tel que qu'au moins un coup a été joué dans $A \parr B$ auparavant alors si le dernier coup dans $A \parr B$ qui précède $c$ est dans $A$ (resp. $B$), on dit que $c$ est dans un $A$-bloc (resp. $B$-bloc). Soit $l$, $r$ une paire d'injections à images disjointes, on note $\bar{s}$ la suite de coups de $A \parr B \multimap C$ obtenue en appliquant $l$ aux coups de $s$ dans $C$ qui sont dans des $A$-blocs et $r$ à ceux qui sont dans des $B$-blocs.

On note $\sigma' = ((p_1;\sigma) \parr (p_2;\sigma)); c_C^{l,r}$, il est facile de voir que $\sigma' = \{\bar{s} \mid s \in \sigma\}$. Il reste à montrer que $\sigma' \approx \sigma$ et pour cela nous allons avoir besoin du fait que $\bar{s} \equiv_{A \parr B \multimap C} s$ ce qui revient à dire que les indices apparaissant dans les $A$-blocs de $s$ sont disjoints de ceux apparaissant dans ses $B$-blocs. Si $s$ ne contient pas de $A$-bloc (resp. $B$-bloc) le résultat est immédiat. Sinon du fait que $C$ est l'interprétation d'une formule de MELLP sans atome, c'est un jeu très bien ouvert (*i.e.* avec un unique premier coup $c_0$) et ses deuxièmes coups sont formés d'un indice et d'un coup fixe $c_1$. Si le même indice $i$ apparaît à la fois dans un $A$-bloc et dans un $B$-bloc, c'est que le coup $(i, c_1)$ apparaît à la fois dans un $A$-bloc et dans un $B$-bloc puisque $s \upharpoonright_C \upharpoonright_i$ est une partie de $C$ par multiplicité. Supposons que la première occurrence est celle dans un $A$-bloc, puisque l'on passe ensuite dans un $B$-bloc c'est qu'il y a entre temps un coup dans $A$ et que le premier coup dans $A \parr B$ après ce $(i, c_1)$ est dans $A$. Par déterminisme, auto-équivalence et absence de mémoire de $\sigma$, si $O$ joue après la deuxième occurrence de $(i, c_1)$ comme après la première alors $J$ doit également jouer pareil ce qui donne une partie de la forme $s_1 a_1 s_2 (i, c_1) s_3 a_2 s_4 b_1 s_5 (i, c_1) s_3 a_2$ (où $s_2$, $s_3$ et $s_5$ sont dans $C$) et contraint $J$ à jouer le même coup $a_2$ dans $A$, ce qui est impossible puisque le coup $b_1$ dans $A \parr B$ qui précède est dans $B$ et on violerait donc la condition de switching du $\parr$.

On peut finalement prouver que $\sigma' \approx \sigma$ ; si $smn^J \in \sigma$, $t \in \sigma'$ et $sm \equiv_{A \parr B \multimap C} tm'$ alors :

- si $n \in M_{A \parr B}$, alors $\overline{sm}n \in \sigma'$ et $\overline{sm} \equiv_{A \parr B \multimap C} sm \equiv_{A \parr B \multimap C} tm'$ donc, puisque $\sigma' \approx \sigma'$, il existe $n'$ tel que $tm'n' \equiv_{A \parr B \multimap C} \overline{sm}n$ et $tm'n' \in \sigma'$ ;
- si $n \in M_C$, soit $n$ est le premier coup de $C$ et le résultat est immédiat, soit $n$ n'est pas initial et puisque $\sigma$ est centrale, $n$ est soit dans un $A$-bloc soit dans un $B$-bloc. On considère le cas d'un $A$-bloc : $\overline{sm}l(n) \in \sigma'$ et $\overline{sm} \equiv_{A \parr B \multimap C} sm \equiv_{A \parr B \multimap C} tm'$ donc, puisque $\sigma' \approx \sigma'$, il existe $n'$ tel que $tm'n' \equiv_{A \parr B \multimap C} \overline{sm}l(n) \equiv_{A \parr B \multimap C} smn$ et $tm'n' \in \sigma'$.

La condition réciproque s'obtient de la même manière. □

7.53. Lemme (!-lemme)
*Soit $\sigma : ?A^\perp \overset{\bullet}{\multimap} ?B_1^\perp \parr \ldots \parr ?B_n^\perp$ une stratégie, on a $\sigma \approx !_A(\uparrow d_A; \sigma \parr A)$.*

Démonstration : Par isomorphisme, on peut écrire $?B_1^\perp \parr \ldots \parr ?B_n^\perp \simeq_a ?(B_1^\perp \oplus \ldots \oplus B_n^\perp)$, on note $\sigma'$ la stratégie obtenue sur $?A^\perp \overset{\bullet}{\multimap} ?(B_1^\perp \oplus \cdots \oplus B_n^\perp)$ et $B^\perp = B_1^\perp \oplus \cdots \oplus B_n^\perp$. Puisque $?A^\perp = \uparrow\flat A^\perp$, par le lemme 7.33, $\sigma' = \downarrow_A \uparrow \sigma'_0$ avec $\sigma'_0 : \sharp B \multimap \sharp A$. Par le lemme 3.39, $\sigma'_0 \approx (\sigma'_0; d_A)^\dagger$ avec $\sigma'_0; d_A : \sharp B \multimap A$. On a alors $\sigma' \approx !_A(\uparrow d_A; (\sigma' \parr A))$ et donc, en appliquant à nouveau l'isomorphisme, $\sigma \approx !_A(\uparrow d_A; (\sigma \parr A))$. □

Pour ne pas rentrer dans des considérations compliquées concernant la notion de stratégie compacte, on va montrer un résultat de définissabilité *locale* des stratégies qui signifie informellement que : de toute stratégie $\sigma$, on peut extraire une dernière règle du calcul des séquents dont l'interprétation a pu mener à $\sigma$. Ceci peut s'exprimer de la manière suivante, où une pseudo-preuve est une preuve potentiellement infinie puisque rien ne garantit que le processus d'extraction d'une dernière règle (que l'on peut appliquer récursivement) termine :

7.54. THÉORÈME (DÉFINISSABILITÉ PROPOSITIONNELLE LOCALE POUR MELLP)
*Soient $A$ et $B$ des formules négatives sans atome de* MELLP *:*

- *Si $\sigma$ est une stratégie centrale gagnante sans mémoire sur $B \multimap A$ alors $\sigma$ appartient à l'interprétation d'une* pseudo-preuve *de* $\vdash B^{\perp}, A$ *dans* MELLP.
- *Si $\sigma$ est une stratégie gagnante sans mémoire sur $A$ alors $\sigma$ appartient à l'interprétation d'une* pseudo-preuve *de* $\vdash A$ *dans* MELLP.

Démonstration : On raisonne de manière très similaire au théorème 7.38.

On ramène tout d'abord le premier cas au deuxième : $B$ s'écrit $?B_1^{\perp} \parr \ldots \parr ?B_n^{\perp}$, si $n > 1$ par le lemme 7.52 on a $\sigma \approx (\sigma_1 \parr \sigma_2); c_C^{l,r}$, ce qui signifie que $\sigma$ correspond à l'interprétation d'un règle $\otimes$ suivie de règles $?c$ appliquées à $\sigma_1$ et $\sigma_2$, auxquelles on applique récursivement le théorème. Si $n = 1$, par le lemme 7.53, $\sigma \approx !_{B_1}\sigma_1$ donc $\sigma$ correspond à l'interprétation d'un règle ! appliquée à $\sigma_1$, à laquelle on applique récursivement le théorème.

Pour le deuxième cas, on note $A = ?A_1^{\perp} \parr \ldots \parr ?A_n^{\perp}$ et puisque $\sigma$ est sans mémoire, $\sigma$ répond toujours le même premier coup joueur $(i, a)$ dans le même sous-jeu $?A_k^{\perp}$. Comme dans la preuve du théorème 7.38, on peut séparer tous les coups joués par $\sigma$ dans $?A_k^{\perp}$ avec l'indice $i$ dans une autre copie de $?A_k^{\perp}$, on obtient ainsi une stratégie $\sigma_1 : ?A_1^{\perp} \parr \ldots \parr ?A_k^{\perp} \parr ?A_k^{\perp} \parr \ldots \parr ?A_n^{\perp}$ telle que d'une part $\sigma$ est obtenue en appliquant à $\sigma_1$ l'interprétation d'une règle de contraction et d'autre part $\sigma_1 = \upharpoonleft\sigma_2$ ce qui signifie qu'elle correspond à l'application de l'interprétation d'une règle $?d$ à $\sigma_2 : A_k \multimap ?A_1^{\perp} \parr \ldots \parr ?A_n^{\perp}$ et on est ramené au premier cas.

Les cas correspondant aux constantes se traitent exactement comme pour le théorème 7.38. □

7.55. REMARQUE : Afin d'étendre ce résultat à LLP, on peut considérer l'approche suivante : si $\pi$ est une preuve de $\vdash N$ dans LLP, on l'interprète par la stratégie associée à la preuve $\pi'$ de $\vdash !N'$ (avec $!N \simeq !N'$) obtenue en appliquant la traduction multiplicative (proposition 4.30) de LLP dans MELLP. Le résultat de définissabilité pour MELLP donne alors la définissabilité pour LLP sans atome : soit $\sigma : !N'$ une stratégie gagnante finie sans mémoire, par le théorème 7.54 il existe une preuve de $\vdash !N'$ dont l'interprétation est $\sigma$. En coupant cette preuve sur celle de $\vdash ?N'^{\perp}, !N$ correspondant à l'isomorphisme $!N \simeq !N'$, on obtient une preuve de $\vdash !N$. Enfin, par réversibilité de la règle ! dans LLP, on obtient une preuve de $\vdash N$.

## 7.4 Liens avec la logique linéaire intuitionniste

### 7.4.1 Des jeux intuitionnistes aux jeux polarisés

Les jeux intuitionnistes (dans lesquels seul $O$ commence) apparaissent comme un cas particulier des jeux polarisés. Or ces jeux intuitionnistes fournissent naturellement un modèle de la logique linéaire intuitionniste alors que les jeux polarisés donnent un modèle de LLP. Via cette interprétation dans les jeux, ILL apparaît donc comme un sous-système de LLP, cette traduction pouvant s'expliciter syntaxiquement.

On peut en particulier voir comment la fermeture cartésienne de la catégorie des jeux intuitionnistes se décompose dans les jeux polarisés :

$$
\begin{array}{lll}
 & (\sharp A \odot \sharp B) \multimap\!\!\!\!\!\!\rightarrow C & \\
\simeq & \downarrow(\sharp A \odot \sharp B) \multimap C & \\
\simeq & (\downarrow\sharp A \otimes \downarrow\sharp B) \multimap C & = (!A \otimes !B) \multimap C \\
\simeq & \downarrow\sharp A \multimap \downarrow\sharp B \multimap C & = !A \multimap !B \multimap C \\
\simeq & \sharp A \multimap\!\!\!\!\!\!\rightarrow \sharp B \multimap\!\!\!\!\!\!\rightarrow C &
\end{array}
$$

tout dépend du choix que l'on fait pour le placement du $\downarrow$, les jeux intuitionnistes l'associent avec $\multimap$ pour donner $\multimap\!\!\!\!\!\!\rightarrow$ alors que les jeux polarisés l'associent avec $\sharp$ pour donner $!$.

Pouvoir se passer des connecteurs $\odot$ et $\sharp$ comme connecteurs logiques résout naturellement les questions de bonne ouverture des jeux puisque ce sont ces deux connecteurs qui sont responsables de la perte de cette propriété.

ILL a été introduit pour étudier la logique intuitionniste dans un cadre plus simple que LL tout entier au prix (cher) d'une perte de la dualité de LL. Il apparaît ici que LLP peut être considéré comme une extension de ILL permettant d'interpréter la logique classique, redonnant sa place à la dualité et possédant cependant les avantages de ILL puisqu'on est en mesure de définir des réseaux de preuve, une sémantique des jeux, ...

### 7.4.2 Jeux de Lamarche

François Lamarche a introduit une notion de jeux [Lam95] qui lui a permis de montrer un résultat de complétude forte pour LL. La base de ce travail est une traduction de LL dans ILL, les jeux sont ensuite utilisés pour interpréter les preuves de ILL. Cependant ces jeux utilisent une notion de décalage qui est exactement celle utilisée ici.

Sans entrer dans une comparaison détaillée entre les jeux polarisés et les jeux de Lamarche, nous allons voir que la traduction de LL dans ILL qu'il a utilisée peut se factoriser par LLP.

On considère un système LLP enrichi avec des décalages $\downarrow$ et $\uparrow$ (dont les règles d'introduction correspondent respectivement à la promotion et à la déréliction) et des connecteurs $\odot$, $\sharp$ et $\flat$ comme cela est suggéré par les jeux. On peut traduire LL sans constantes additives dans ce système par :

$$
\begin{array}{ccc}
X & \rightsquigarrow & \uparrow X \\
A \otimes B & \rightsquigarrow & \downarrow(A \odot B) \\
A \oplus B & \rightsquigarrow & \uparrow(A \oplus B) \\
1 & \rightsquigarrow & \downarrow\top \\
!A & \rightsquigarrow & \downarrow\sharp A
\end{array}
$$

en ajoutant les décalages supplémentaires nécessaires à la bonne polarisation des formules.

Le séquent $\vdash \Gamma$ est traduit formule par formule puis en ajoutant les $\uparrow$ nécessaires en tête pour rendre toutes les formules négatives.

L'image de LL peut alors être traduite dans ILL en utilisant le fait que, d'après les jeux, les connecteurs de ILL sont les connecteurs négatifs de LLP :

$$
\begin{array}{ccc}
X & \rightsquigarrow & X \\
A \odot B & \rightsquigarrow & A \otimes B \\
A \,\&\, B & \rightsquigarrow & A \,\&\, B \\
\top & \rightsquigarrow & 1 \\
\sharp A & \rightsquigarrow & !A
\end{array}
$$

Le séquent $\vdash \mathcal{N}$ est traduit par $\overline{\mathcal{N}^\star}^\perp \vdash$ où $\mathcal{N}^\star$ est la traduction des formules de $\mathcal{N}$ et l'opération $\overline{(.)}$ consiste à supprimer le $\uparrow$ de tête.

On obtient ainsi la traduction de Lamarche.

7.56. Exemple
Nous reprenons l'exemple donné dans [Lam95]. Soient $\alpha$ et $\beta$ des atomes négatifs,

$$\begin{array}{cl}
(\alpha \oplus \beta^\perp) \otimes (?\beta \,\&\, \perp) & \text{formule initiale} \\
\downarrow(\uparrow(\alpha \oplus \uparrow\beta^\perp) \odot \downarrow(\uparrow\flat\beta \,\&\, \uparrow 0)) & \text{traduction brute dans } \mathsf{LLP} \\
\downarrow(\uparrow(\downarrow\alpha \oplus \downarrow\uparrow\beta^\perp) \odot \uparrow\downarrow(\uparrow\flat\downarrow\beta \,\&\, \uparrow 0)) & \text{ajout des décalages nécessaires} \\
\downarrow(\uparrow(\downarrow\alpha \oplus \downarrow\uparrow\beta^\perp) \otimes \uparrow\downarrow(\uparrow ?\downarrow\beta \,\&\, \uparrow\perp)) & \text{traduction dans } \mathsf{ILL}
\end{array}$$

ce qui donne bien le même résultat que la traduction de Lamarche au $\downarrow$ extérieur près.

# Chapitre 8

# Géométrie de l'interaction

La GoI initialement introduite pour MELL [Gir88] a ensuite été étendue à LL tout entier [Gir95a]. Cependant la solution est complexe en particulier en ce qui concerne l'interaction entre connecteurs additifs et connecteurs exponentiels. Avant d'aborder la question de la GoI polarisée nous allons revenir sur l'interprétation des connecteurs additifs afin de simplifier la solution donnée par Girard [Gir95a] et d'en donner une version plus parallèle et conforme à l'idée de réseaux.

## 8.1 Interprétation des connecteurs additifs

Nous reprenons une partie des résultats de [Lau01] qui fournissent une interprétation en GoI de MALL, ces résultats sont basés sur des idées provenant de [Gir95a] en particulier en ce qui concerne le système $\mathsf{LL}^\flat$.

Le principal problème lié à l'interprétation des connecteurs additifs en GoI provient de la duplication et de l'effacement contenus dans les étapes d'élimination des coupures additives, par exemple dans :

$$\dfrac{\dfrac{\dfrac{\pi_1}{\vdash \Gamma, A} \quad \dfrac{\pi_2}{\vdash \Gamma, B}}{\vdash \Gamma, A \,\&\, B}\,\& \qquad \dfrac{\dfrac{\pi_3}{\vdash \Delta, A^\perp}}{\vdash \Delta, A^\perp \oplus B^\perp}\,\oplus_1}{\vdash \Gamma, \Delta}\,cut$$

$$\downarrow$$

$$\dfrac{\dfrac{\pi_1}{\vdash \Gamma, A} \quad \dfrac{\pi_3}{\vdash \Delta, A^\perp}}{\vdash \Gamma, \Delta}\,cut$$

la preuve $\pi_2$ est effacée. La parfaite linéarité de la GoI n'est pas directement compatible avec un tel effacement.

### 8.1.1 Le système $\mathsf{MALL}^\flat$

On ajoute un nouveau symbole $\flat$ qui permet de marquer certains séquents comme «partiels». On aura donc des séquents de la forme $\vdash \Gamma$ et des séquents de la forme $\vdash \Gamma, \flat$ signifiant qu'une partie de la preuve peut manquer. Ce symbole n'est pas une formule et aucun connecteur ne pourra lui être appliqué.

Le système $\mathsf{MALL}^\flat$ est obtenu en ajoutant les deux règles $\flat$ et $s^\flat$ à MALL (si nous redonnons les règles de tout le système c'est parce que nous y introduisons des indices sur les formules qui ne sont absolument pas à prendre en compte pour le système mais nous permettrons de distinguer les occurrences de formules dans les règles pour donner les transitions de la IAM) :

$$\frac{}{\vdash A, A^{\perp}}\,ax \qquad \frac{\vdash \Gamma_1, A \quad \vdash A^{\perp}, \Delta_1}{\vdash \Gamma, \Delta}\,cut$$

$$\frac{\vdash \Gamma_1, A \quad \vdash \Delta_1, B}{\vdash \Gamma, \Delta, A \otimes B}\,\otimes \qquad \frac{\vdash \Gamma_1, A, B}{\vdash \Gamma, A \parr B}\,\parr$$

$$\frac{\vdash \Gamma_1, A \quad \vdash \Gamma_2, B}{\vdash \Gamma, A \,\&\, B}\,\& \qquad \frac{\vdash \Gamma_1, A}{\vdash \Gamma, A \oplus B}\,\oplus_1 \qquad \frac{\vdash \Gamma_1, B}{\vdash \Gamma, A \oplus B}\,\oplus_2$$

$$\frac{}{\vdash \Gamma, \flat}\,\flat \qquad \frac{\vdash \Gamma_1, \Delta_1 \quad \vdash \Gamma_2, \flat \quad \vdash \Delta_2, \flat}{\vdash \Gamma, \Delta}\,s^{\flat}$$

La règle $\flat$ montre que les séquents $\vdash \Gamma, \flat$ correspondent à des preuves partielles.

Les deux prémisses de la règle $s^{\flat}$ qui sont marquées avec le symbole $\flat$ vont nous permettre de modifier l'élimination des coupures de manière à la rendre linéaire au sens de la GoI.

L'élimination des coupures pour $\mathsf{MALL}^{\flat}$ est définie comme pour $\mathsf{MALL}$ excepté pour le cas de la coupure $\&/\oplus_i$ :

$$\dfrac{\dfrac{\dfrac{\pi_1}{\vdash \Gamma_1, A} \quad \dfrac{\pi_2}{\vdash \Gamma_2, B}}{\vdash \Gamma_3, A \,\&\, B}\,\& \quad \dfrac{\dfrac{\pi_3}{\vdash \Delta_1, A^{\perp}}}{\vdash \Delta_2, A^{\perp} \oplus B^{\perp}}\,\oplus_1}{\vdash \Gamma, \Delta}\,cut$$

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{\pi_1}{\vdash \Gamma_1, A} \quad \dfrac{\pi_3^1}{\vdash \Delta_1^1, A^{\perp}}}{\vdash \Gamma_3, \Delta_2}\,cut \quad \dfrac{\dfrac{\pi_2}{\vdash \Gamma_2, B} \quad \dfrac{}{\vdash B^{\perp}, \flat}\,\flat}{\vdash \Gamma_4, \flat}\,cut \quad \dfrac{\dfrac{\pi_3^2}{\vdash \Delta_1^2, A^{\perp}} \quad \dfrac{}{\vdash A, \flat}\,\flat}{\vdash \Delta_3, \flat}\,cut}{\vdash \Gamma, \Delta}\,s^{\flat}$$

À cause de cette nouvelle étape d'élimination des coupures, $\mathsf{MALL}$ n'est pas un sous-système de $\mathsf{MALL}^{\flat}$ stable par réduction.

Nous devons également définir de nouvelles étapes commutatives de réduction pour la règle $s^{\flat}$ :

$$\dfrac{\dfrac{\dfrac{\pi_1}{\vdash \Gamma_1, \Delta_1, C_1} \quad \dfrac{\pi_2}{\vdash \Gamma_2, C_2, \flat} \quad \dfrac{\pi_3}{\vdash \Delta_2, \flat}}{\vdash \Gamma_3, \Delta_3, C}\,s^{\flat} \quad \dfrac{\pi_4}{\vdash \Sigma_1, C^{\perp}}}{\vdash \Gamma, \Delta, \Sigma}\,cut$$

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{\pi_1}{\vdash \Gamma_1, \Delta_1, C_1} \quad \dfrac{\pi_4^1}{\vdash \Sigma_1, C_1^{\perp}}}{\vdash \Gamma_3, \Delta_3, \Sigma_3}\,cut \quad \dfrac{\dfrac{\pi_2}{\vdash \Gamma_2, C_2, \flat} \quad \dfrac{\pi_4^2}{\vdash \Sigma_2, C_2^{\perp}}}{\vdash \Gamma_4, \Sigma_4, \flat}\,cut \quad \dfrac{\pi_3}{\vdash \Delta_2, \flat}}{\vdash \Gamma, \Delta, \Sigma}\,s^{\flat}$$

et de même si la formule coupée est dans $\Delta$.

8.1. Définition (Quasi forme normale)
Une preuve de $\mathsf{MALL}^{\flat}$ est dite en *quasi forme normale* si elle ne peut pas être réduite par une des étapes ci-dessus.

8.2. Lemme (Coupures d'une quasi forme normale)
*Soit $\pi$ une preuve de $\mathsf{MALL}^\flat$ en quasi forme normale et $c$ une coupure présente dans $\pi$, au moins une des formules coupées a été introduite par une règle $\flat$ et n'a ensuite été contexte que de coupures.*

Démonstration : Tout d'abord si le séquent conclusion de $c$ ne contient pas $\flat$, $c$ peut être réduite par une étape d'élimination des coupures de $\mathsf{MALL}$, donc $\pi$ n'est pas en quasi forme normale. Le séquent conclusion de $c$ est donc de la forme $\vdash \Gamma, \Delta, \flat$ et on suppose que $c$ a une prémisse $\vdash \Gamma, A, \flat$ où $A$ est la formule coupée. On montre le résultat par récurrence sur le nombre de coupures situées au-dessus de $c$ :

- S'il n'y en a aucune, soit la règle qui introduit $\vdash \Gamma, A, \flat$ est une règle $\flat$ et le résultat est prouvé soit c'est une autre règle et $c$ peut être réduite donc $\pi$ n'est pas en quasi forme normale.
- S'il y en a $n+1$, soit la règle qui introduit $\vdash \Gamma, A, \flat$ est une coupure et le résultat est prouvé par hypothèse de récurrence appliquée à cette coupure, soit c'est une autre règle et $\pi$ n'est pas en quasi forme normale. $\square$

Il serait possible de définir (comme c'est le cas dans [Gir95a]) une procédure complète d'élimination des coupures pour $\mathsf{MALL}^\flat$, toutefois ceci nous imposerait de nombreuses étapes commutatives et compliquerait les choses inutilement. En effet, nous nous intéressons en fait à la normalisation de $\mathsf{MALL}$ et pour ce faire l'élimination partielle des coupures de $\mathsf{MALL}^\flat$ et l'existence des quasi formes normales suffit grâce à la traduction $(.)^\natural$ :

8.3. Définition (Traduction $(.)^\natural$)
Soit $\pi$ une preuve de $\mathsf{MALL}^\flat$ dont la conclusion $\vdash \Gamma$ ne contient pas $\flat$, la preuve $\pi^\natural$ de $\vdash \Gamma$ dans $\mathsf{MALL}$ est obtenue en remplaçant toute sous-preuve de $\pi$ d'un séquent sans $\flat$ terminant par une règle $s^\flat$ par la preuve de sa prémisse sans $\flat$.

Comme nous nous intéressons aux preuves de $\mathsf{MALL}$, celles de $\mathsf{MALL}^\flat$ n'étant qu'un outil, il est naturel de se restreindre au sous-système de $\mathsf{MALL}^\flat$ dont les preuves ne contiennent pas de $\flat$ en conclusion. Ce sous-système est, bien entendu, stable par élimination des coupures et contient les réduits des preuves de $\mathsf{MALL}$.

8.4. Lemme (Traduction $(.)^\natural$ et forme normale)
*Soit $\pi$ une preuve de $\mathsf{MALL}^\flat$ en quasi forme normale dont la conclusion ne contient pas $\flat$, $\pi^\natural$ est une preuve sans coupure de $\mathsf{MALL}$.*

Démonstration : D'après le lemme 8.2, toute coupure dans $\pi$ a une conclusion qui contient $\flat$ et, puisque la traduction $(.)^\natural$ supprime toutes les sous-preuves de tels séquents, les coupures de $\pi$ sont effacées dans $\pi^\natural$. $\square$

8.5. Remarque : Si $\pi$ est une preuve de $\mathsf{MALL}$ et $\pi_0$ est une quasi forme normale de $\pi$ obtenue par la réduction de $\mathsf{MALL}^\flat$, alors $\pi_0^\natural$ est une forme normale de $\pi$ par réduction de $\mathsf{MALL}$.

Tout comme les règles & permettent de définir les tranches additives dans $\mathsf{LL}$, on peut étendre cette notion en présence de règles $s^\flat$ en considérant qu'une tranche est obtenue en ne gardant qu'une prémisse des règles & et soit la prémisse $\vdash \Gamma, \Delta$ soit les deux prémisses $\vdash \Gamma, \flat$ et $\vdash \Delta, \flat$ pour chaque règle $s^\flat$.

Nous allons désormais associer une notion de poids aux preuves de $\mathsf{MALL}^\flat$ comme nous l'avons fait dans les réseaux en suivant [Gir96]. La notion de poids que l'on va utiliser ici est celle introduite à la section 5.2 et sert toujours à décrire les tranches additives. Les notions de valuation et de dépendance restent les mêmes, on note $w(p)$ un poids $w \leq p$, $w(\bar{p})$ un poids $w \leq \bar{p}$ et $w(\not p)$ un poids $w$ qui ne dépend pas de $p$.

8.6. Définition (Poids littéral)
Un *poids littéral* est soit un poids élémentaire $p$ soit une négation de poids élémentaire $\bar{p}$.

À chaque règle & et chaque règle $s^\flat$ est associé un poids littéral, appelé *poids propre* de la règle. Ceci généralise très légèrement le cas des réseaux où l'on associait toujours un poids élémentaire aux nœuds &. De plus on ne va plus imposer que ces poids soient tous distincts mais on utilise une contrainte plus faible :

8.7. Définition (Pondération correcte)
Une preuve $\pi$ de $\mathsf{MALL}^\flat$ est dite *pondérée* si un poids littéral est associé à chaque règle & et $s^\flat$ de $\pi$.

Une telle pondération est *correcte* si, lorsque deux règles ont des poids propres correspondant au même poids élémentaire, elles se trouvent au-dessus de deux prémisses différentes d'une règle & ou bien l'une au-dessus de la prémisse $\vdash \Gamma, \Delta$ d'une règle $s^\flat$ et l'autre au-dessus d'une de ses prémisses $\flat$. Autrement dit, un poids élémentaire ne peut correspondre qu'à une seule règle par tranche additive.

Si $\pi$ est une preuve pondérée qui se réduit en $\pi'$ par une étape d'élimination des coupures, on associe de manière immédiate une pondération à $\pi'$, avec les deux cas particuliers suivants :
- dans le cas d'une étape $\&/\oplus_i$, la règle $s^\flat$ obtenue a le même poids propre que la règle & de départ si la règle $\oplus_i$ est une règle $\oplus_1$ et sa négation si c'est une règle $\oplus_2$,
- si une preuve est dupliquée, les deux copies sont pondérées de la même manière.

Si la pondération de $\pi$ est correcte, il est facile de voir que celle de $\pi'$ également. On va désormais considérer que les preuves sont correctement pondérées.

8.8. Remarque : Si on considère que chaque sous-preuve est munie du poids de la tranche à laquelle elle appartient, on peut donner un sens au fait que la nouvelle étape de réduction $\&/\oplus_i$ est véritablement linéaire. En effet, avant réduction on a $p.\pi_1 + \bar{p}.\pi_2 + \pi_3$ où $p$ est le poids propre de la règle & et après réduction on obtient $p.\pi_1 + \bar{p}.\pi_2 + p.\pi_3^1 + \bar{p}.\pi_3^2$, c'est pour cette raison que l'on a introduit la règle $s^\flat$ à trois prémisses (deux prémisses comme dans [Gir95a] n'étant pas suffisantes pour cela).

### 8.1.2 La machine à jetons

On peut désormais étendre la *IAM* aux connecteurs additifs. Cette extension ne modifie pas la partie multiplicative des jetons et en oubliant toute information additive, on retrouve la machine pour $\mathsf{MLL}$.

8.9. Définition (Jeton)
Pour le cas multiplicatif-additif, un *jeton* est un triplet $(m, a, w)$ où $m$ et $a$ sont des piles construites à partir des symboles $g$ et $d$ et $w$ est un monôme ce qui entraîne que l'on a toujours $w(p)$, $w(\bar{p})$ ou $w(\not{p})$.

8.10. Définition (Machine abstraite)
Un *état* de la machine est défini comme dans le cas de $\mathsf{MELL}$ (définition 2.12) en utilisant des jetons $(m, a, w)$.

Les transitions de la machine sont définies pour chaque règle de la manière suivante, où $\Gamma$ (resp. $\Delta$) signifie que la formule de l'état avant la transition est dans le multi-ensemble $\Gamma$ (resp. $\Delta$) et que la formule de l'état après la transition est la même.

$$
\begin{array}{lcl}
\multicolumn{3}{c}{ax} \\
A^{\uparrow}(m,a,w) & \rightarrow & A^{\perp\downarrow}(m,a,w) \\
A^{\perp\uparrow}(m,a,w) & \rightarrow & A^{\downarrow}(m,a,w)
\end{array}
$$

$$
\begin{array}{lcl}
\multicolumn{3}{c}{cut} \\
A^{\downarrow}(m,a,w) & \rightarrow & A^{\perp\uparrow}(m,a,w) \\
A^{\perp\downarrow}(m,a,w) & \rightarrow & A^{\uparrow}(m,a,w) \\
\Gamma^{\uparrow}(m,a,w) & \rightarrow & \Gamma_1^{\uparrow}(m,a,w) \\
\Delta^{\uparrow}(m,a,w) & \rightarrow & \Delta_1^{\uparrow}(m,a,w) \\
\Gamma_1^{\downarrow}(m,a,w) & \rightarrow & \Gamma^{\downarrow}(m,a,w) \\
\Delta_1^{\downarrow}(m,a,w) & \rightarrow & \Delta^{\downarrow}(m,a,w)
\end{array}
$$

$$
\begin{array}{lcl}
\multicolumn{3}{c}{\otimes} \\
A \otimes B^{\uparrow}(g.m,a,w) & \rightarrow & A^{\uparrow}(m,a,w) \\
A \otimes B^{\uparrow}(d.m,a,w) & \rightarrow & B^{\uparrow}(m,a,w) \\
A \otimes B^{\uparrow}(\varepsilon,a,w) & \rightarrow & \emptyset \\
A^{\downarrow}(m,a,w) & \rightarrow & A \otimes B^{\downarrow}(g.m,a,w) \\
B^{\downarrow}(m,a,w) & \rightarrow & A \otimes B^{\downarrow}(d.m,a,w) \\
\Gamma^{\uparrow}(m,a,w) & \rightarrow & \Gamma_1^{\uparrow}(m,a,w) \\
\Delta^{\uparrow}(m,a,w) & \rightarrow & \Delta_1^{\uparrow}(m,a,w) \\
\Gamma_1^{\downarrow}(m,a,w) & \rightarrow & \Gamma^{\downarrow}(m,a,w) \\
\Delta_1^{\downarrow}(m,a,w) & \rightarrow & \Delta^{\downarrow}(m,a,w)
\end{array}
$$

$$
\begin{array}{lcl}
\multicolumn{3}{c}{\parr} \\
A \parr B^{\uparrow}(g.m,a,w) & \rightarrow & A^{\uparrow}(m,a,w) \\
A \parr B^{\uparrow}(d.m,a,w) & \rightarrow & B^{\uparrow}(m,a,w) \\
A \parr B^{\uparrow}(\varepsilon,a,w) & \rightarrow & \emptyset \\
A^{\downarrow}(m,a,w) & \rightarrow & A \parr B^{\downarrow}(g.m,a,w) \\
B^{\downarrow}(m,a,w) & \rightarrow & A \parr B^{\downarrow}(d.m,a,w) \\
\Gamma^{\uparrow}(m,a,w) & \rightarrow & \Gamma_1^{\uparrow}(m,a,w) \\
\Gamma_1^{\downarrow}(m,a,w) & \rightarrow & \Gamma^{\downarrow}(m,a,w)
\end{array}
$$

$$
\begin{array}{lcl}
\multicolumn{3}{c}{\&} \\
A \,\&\, B^{\uparrow}(m,g.a,w(p)) & \rightarrow & A^{\uparrow}(m,a,w(p)) \\
A \,\&\, B^{\uparrow}(m,g.a,w(\not p)) & \rightarrow & A^{\uparrow}(m,a,w.p) \\
A \,\&\, B^{\uparrow}(m,g.a,w(\bar{p})) & \rightarrow & \emptyset \\
A \,\&\, B^{\uparrow}(m,d.a,w(\bar{p})) & \rightarrow & B^{\uparrow}(m,a,w(\bar{p})) \\
A \,\&\, B^{\uparrow}(m,d.a,w(\not p)) & \rightarrow & B^{\uparrow}(m,a,w.\bar{p}) \\
A \,\&\, B^{\uparrow}(m,d.a,w(p)) & \rightarrow & \emptyset \\
A \,\&\, B^{\uparrow}(m,\varepsilon,w) & \rightarrow & \emptyset \\
A^{\downarrow}(m,a,w) & \rightarrow & A \,\&\, B^{\downarrow}(m,g.a,w.p) \\
B^{\downarrow}(m,a,w) & \rightarrow & A \,\&\, B^{\downarrow}(m,d.a,w.\bar{p}) \\
\Gamma^{\uparrow}(m,a,w(p)) & \rightarrow & \Gamma_1^{\uparrow}(m,a,w(p)) \\
\Gamma^{\uparrow}(m,a,w(\bar{p})) & \rightarrow & \Gamma_2^{\uparrow}(m,a,w(\bar{p})) \\
\Gamma^{\uparrow}(m,a,w(\not p)) & \rightarrow & \emptyset \\
\Gamma_1^{\downarrow}(m,a,w) & \rightarrow & \Gamma^{\downarrow}(m,a,w.p) \\
\Gamma_2^{\downarrow}(m,a,w) & \rightarrow & \Gamma^{\downarrow}(m,a,w.\bar{p})
\end{array}
$$

$$
\begin{array}{lcl}
\multicolumn{3}{c}{\flat} \\
\Gamma^{\uparrow}(m,a,w) & \rightarrow & \emptyset
\end{array}
$$

$$
\begin{array}{lcl}
\multicolumn{3}{c}{\oplus_1} \\
A \oplus B^{\uparrow}(m,g.a,w) & \rightarrow & A^{\uparrow}(m,a,w) \\
A \oplus B^{\uparrow}(m,d.a,w) & \rightarrow & \emptyset \\
A \oplus B^{\uparrow}(m,\varepsilon,w) & \rightarrow & \emptyset \\
A^{\downarrow}(m,a,w) & \rightarrow & A \oplus B^{\downarrow}(m,g.a,w) \\
\Gamma^{\uparrow}(m,a,w) & \rightarrow & \Gamma_1^{\uparrow}(m,a,w) \\
\Gamma_1^{\downarrow}(m,a,w) & \rightarrow & \Gamma^{\downarrow}(m,a,w)
\end{array}
$$

$$
\begin{array}{lcl}
\multicolumn{3}{c}{s^{\flat}} \\
\Gamma^{\uparrow}(m,a,w(p)) & \rightarrow & \Gamma_1^{\uparrow}(m,a,w(p)) \\
\Gamma^{\uparrow}(m,a,w(\bar{p})) & \rightarrow & \Gamma_2^{\uparrow}(m,a,w(\bar{p})) \\
\Gamma^{\uparrow}(m,a,w(\not p)) & \rightarrow & \emptyset \\
\Delta^{\uparrow}(m,a,w(p)) & \rightarrow & \Delta_1^{\uparrow}(m,a,w(p)) \\
\Delta^{\uparrow}(m,a,w(\bar{p})) & \rightarrow & \Delta_2^{\uparrow}(m,a,w(\bar{p})) \\
\Delta^{\uparrow}(m,a,w(\not p)) & \rightarrow & \emptyset \\
\Gamma_1^{\downarrow}(m,a,w) & \rightarrow & \Gamma^{\downarrow}(m,a,w.p) \\
\Gamma_2^{\downarrow}(m,a,w) & \rightarrow & \Gamma^{\downarrow}(m,a,w.\bar{p}) \\
\Delta_1^{\downarrow}(m,a,w) & \rightarrow & \Delta^{\downarrow}(m,a,w.p) \\
\Delta_2^{\downarrow}(m,a,w) & \rightarrow & \Delta^{\downarrow}(m,a,w.\bar{p})
\end{array}
$$

$p$ est le poids propre la règle & ou de la règle $s^{\flat}$ et si le résultat d'une transition est tel que $w = 0$, on considère que c'est l'état $\emptyset$.

Les transitions pour la règle $\oplus_2$ sont définies de la manière évidente à partir de celles pour la règle $\oplus_1$.

8.11. Remarque : On peut modifier les transitions de la règle & par :

$$\begin{array}{rcl} A \,\&\, B^{\uparrow}(m, g.a, w(\not p)) & \rightarrow & A^{\uparrow}(m, a, w(\not p)) \\ A \,\&\, B^{\uparrow}(m, d.a, w(\not p)) & \rightarrow & B^{\uparrow}(m, a, w(\not p)) \end{array}$$

ceci ne change aucun des résultats à venir puisque l'information sur $p$ sera ajoutée dans tous les cas lorsque le jeton traversera la règle & vers le bas.

De la même manière on pourrait modifier les transitions descendantes dans les contextes des règles & et $s^\flat$ pour qu'elles ne modifient pas le poids.

Ce que l'on cherche à obtenir est que le poids du jeton soit toujours plus petit que celui de la tranche dans laquelle il se trouve.

8.12. Définition (Fonction d'interaction)
Soit $\pi$ une preuve, $A$ une conclusion de $\pi$, et $(m, a, w)$ un jeton, on définit une fonction partielle $f_\pi$, appelée *fonction d'interaction*, associée à $\pi$ selon le résultat du calcul de la machine à jetons à partir de l'état $A^{\uparrow}(m, a, w)$, par :

$$f_\pi(A, (m, a, w)) = \begin{cases} (B, (m', a', w')) & \text{si le calcul de la machine termine dans l'état } B^{\downarrow}(m', a', w') \\ \emptyset & \text{si le calcul s'arrête dans l'état } \emptyset \\ \uparrow & \text{si le calcul ne termine pas (calcul infini)} \end{cases}$$

Notation : Si $j$ est un jeton $(m, a, w)$ et $w_0$ est un poids, on note $j.w_0$ le jeton $(m, a, w.w_0)$ et, de même, si $e$ est une paire (formule,jeton) $(A, j)$, on note $e.w_0 = (A, j.w_0)$.

8.13. Lemme (Monotonie par rapport au poids)
*Soit $\pi$ une preuve, $e$ et $e'$ deux paires (formule conclusion,jeton) et $p$ un poids littéral n'apparaissant pas dans $\pi$ tels que le poids $w$ de $e$ vérifie $w.p \neq 0$,*

$$\begin{gathered} f_\pi(e) = \uparrow \Rightarrow f_\pi(e.p) = \uparrow \\ f_\pi(e) = e' \Rightarrow f_\pi(e.p) = e'.p \end{gathered}$$

Démonstration : On montre que, partant de l'état correspondant à $e.p$, le calcul de la machine passe exactement par les mêmes états (avec un poids multiplié par $p$) qu'en partant de $e$. Pour cela il suffit de raisonner par récurrence sur la longueur de l'exécution, en montrant que cette propriété est préservée par toutes les transitions. □

8.14. Théorème (Correction)
*Si $\pi$ est une preuve de $\mathsf{MALL}^\flat$ et $\pi_0$ est une quasi forme normale de $\pi$, alors pour toute paire (formule conclusion,jeton) $e$ :*
- *si $f_\pi(e) = \emptyset$ alors $f_{\pi_0}(e) = \emptyset$,*
- *si $f_{\pi_0}(e) = \uparrow$ alors $f_\pi(e) = \uparrow$,*
- *si $f_{\pi_0}(e) = e'$ alors $f_\pi(e) = e'$,*
- *si $f_\pi(e) = \uparrow$ et $f_{\pi_0}(e) = \emptyset$ alors il existe un poids $w_0$ tel que $f_{\pi_0}(e.w_0) = \uparrow$.*
- *si $f_\pi(e) = e'$ et $f_{\pi_0}(e) = \emptyset$ alors il existe un poids $w_0$ tel que $f_{\pi_0}(e.w_0) = e'.w_0 = f_\pi(e.w_0)$.*

La nécessité d'introduire le paramètre $w_0$ vient du fait que dans l'étape d'élimination des coupures $\&/\oplus_i$, la preuve $\pi_3$ est séparée en deux copies. Or l'équation $\pi_3 = p.\pi_3 + \bar{p}.\pi_3$ ne correspond pas à une identité pour la machine qui peut avoir besoin de connaître de l'information sur $p$ dans le second cas qui n'était pas nécessaire dans le premier.

Démonstration : Il suffit de montrer que, pour chaque étape d'élimination des coupures, le théorème est vérifié et on conclut par récurrence sur la longueur de la réduction de $\pi$ à $\pi_0$.

On suppose que la coupure qui est éliminée est la dernière règle de $\pi$ et on note $\pi'$ la preuve obtenue. Si ce n'est pas le cas, il suffit de remarquer qu'ajouter les mêmes règles à $\pi$ et à $\pi'$ préserve le théorème.

On utilise la notation $e = (\Gamma, j)$ ou $s = \Gamma^{\uparrow}(m, a, w)$ lorsque la formule considérée est dans le multi-ensemble $\Gamma$ mais désormais il ne s'agit plus nécessairement de la même formule avant et après le calcul.

On ne considère que le cas de la coupure $\&/\oplus_i$ qui est de loin le plus important et qui comporte d'une part l'idée de la preuve pour les autres cas et d'autre part toutes les difficultés.

Soit $p$ le poids propre de la règle &, on considère chaque cas possible pour $e$. Les indices et notations correspondent à ceux donnés dans la règle de réduction page 118.

- Si $e = (\Gamma, (m, a, w(p)))$, on considère la suite $s_1$, $s_2$, ... (resp. $s'_1$, $s'_2$, ...) des états $F^{\uparrow}(m, a, w)$ atteints par la machine $M_\pi$ (resp. $M_{\pi'}$) et où $F$ est dans les conclusions des sous-preuves $\pi_1$, $\pi_2$ et $\pi_3$ (resp. $\pi_1$, $\pi_2$, $\pi_3^1$ et $\pi_3^2$).

  En fait ces états sont tous dans les conclusions de $\pi_1$ et $\pi_3$ (resp. $\pi_1$ et $\pi_3^1$) avec $s_1$ dans $\Gamma_1$, plus précisément :
  - Si $s_{2i+1} = F^{\uparrow}(m, a, w)$ avec $F = \Gamma_1$ ou $A$ alors soit $s_{2i+2}$ n'existe pas (le calcul s'arrête avant), soit $s_{2i+2} = A^{\perp\uparrow}(m', a', w')$ avec $f_{\pi_1}(F, (m, a, w)) = (A, (m', a', w'))$ ;
  - Si $s_{2i} = A^{\perp\uparrow}(m, a, w)$ alors soit $s_{2i+1}$ n'existe pas, soit $s_{2i+1} = A^{\uparrow}(m', a', w')$ avec $f_{\pi_3}(A^{\perp}, (m, a, w)) = (A^{\perp}, (m', a', w'))$.

  On a exactement la même propriété pour les états $s'_i$ en remplaçant $\pi_3$ par $\pi_3^1$, d'où $\forall i, s_i = s'_i$.

  Si la suite $s_1$, $s_2$, ... est infinie alors la suite $s'_1$, $s'_2$, ... également. Si $s_1$, $s_2$, ... est finie, ce peut être pour trois raisons différentes. Si $s_n$, dans les conclusions de $\pi_k$, est le dernier élément de cette suite, soit le calcul de la machine $M_{\pi_k}$ à partir de cet état est infini et n'atteint jamais de nouvel état montant dans les conclusions de $\pi_1$ et $\pi_3$, soit $M_{\pi_k}$ atteint l'état $\emptyset$, soit le calcul de $M_{\pi_k}$ à partir de $s_n$ donne un état descendant dans le contexte de $\pi_k$ et la machine $M_\pi$ rend un résultat dans la conclusion de $\pi$. Dans chacun de ces trois cas, la même chose se produit pour les $s'_i$.
- Si $e = (\Gamma, (m, a, w(\bar{p})))$, soit $f_{\pi_2}(e) = (\Gamma, j')$ et dans ce cas $f_\pi(e) = (\Gamma, j') = f_{\pi'}(e)$, soit $f_{\pi_2}(e) = (B, j')$ et alors $f_\pi(e) = \uparrow = f_{\pi'}(e)$. Reste les deux cas $f_{\pi_2}(e) = \emptyset = f_\pi(e) = f_{\pi'}(e)$ et $f_{\pi_2}(e) = \uparrow = f_\pi(e) = f_{\pi'}(e)$.
- Si $e = (\Gamma, (m, a, w(\not{p})))$ alors $f_\pi(e) = \uparrow$ et $f_{\pi'}(e) = \uparrow$.
- Si $e = (\Delta, (m, a, w(p)))$, comme dans le cas $(\Gamma, (m, a, w(p)))$.
- Si $e = (\Delta, (m, a, w(\bar{p})))$, soit $f_{\pi_3}(e) = (\Delta, j')$ et dans ce cas $f_\pi(e) = (\Delta, j') = f_{\pi'}(e)$, soit $f_{\pi_3}(e) = (A^{\perp}, j')$ et dans ce cas $f_\pi(e) = \uparrow = f_{\pi'}(e)$, et sinon $f_{\pi_3}(e) = \emptyset = f_\pi(e) = f_{\pi'}(e)$ ou $f_{\pi_3}(e) = \uparrow = f_\pi(e) = f_{\pi'}(e)$.
- Si $e = (\Delta, (m, a, w(\not{p})))$ alors $f_{\pi'}(e) = \emptyset$ mais on peut avoir $f_\pi(e) = (F, (m', a', w'))$ ou $f_\pi(e) = \uparrow$. Dans ce cas, puisque la pondération est correcte et puisque $w$ ne dépend pas de $p$, par le lemme 8.13 en appliquant le cas $e = (\Delta, (m, a, w(p)))$, on a $f_{\pi'}(\Delta, (m, a, w.p)) = (F, (m', a', w'.p)) = f_\pi(\Delta, (m, a, w.p))$ ou $f_{\pi'}(\Delta, (m, a, w.p)) = \uparrow = f_\pi(\Delta, (m, a, w.p))$. Ce cas est extrêmement important puisqu'il montre pourquoi $f_\pi$ et $f_{\pi'}$ peuvent différer. □

8.15. Corollaire (Correction avec poids saturé)
*Si $w$ est un monôme qui contient $p$ ou $\bar{p}$ pour tout poids propre $p$ apparaissant dans la preuve $\pi$ alors $f_\pi = f_{\pi_0}$.*

Démonstration : Le résultat est immédiat à partir du théorème puisque le seul cas où $f_\pi$ et $f_{\pi_0}$ peuvent différer est $f_\pi(e) = e'$ ou $\uparrow$ et $f_{\pi_0}(e) = \emptyset$ mais dans ce cas $f_\pi(e.w_0) = f_{\pi_0}(e.w_0)$ or $ww_0 = w$. □

8.16. Théorème (Terminaison)
*Soit $\pi$ une preuve et $e$ une paire (formule conclusion,jeton), le calcul de la machine à partir de l'état initial correspondant à $e$ termine toujours, autrement dit $f_\pi(e)$ est une fonction totale.*

Démonstration : Soit $\pi_0$ une quasi forme normale de $\pi$. Par le théorème 8.14, il suffit de montrer que le calcul dans $\pi_0$ termine toujours.

Le chemin parcouru par le jeton, lors d'un calcul de la machine (si elle n'atteint pas l'état $\emptyset$), dans une quasi forme normale, est le suivant : le jeton monte jusqu'à un axiome puis descend jusqu'à une coupure, remonte jusqu'à un axiome, ... Pour que le calcul soit infini, il faut donc que le jeton n'atteigne jamais de conclusion et pour cela il doit emprunter au moins une coupure mais si cela se produit dans une quasi forme normale, par le lemme 8.2, le jeton remonte ensuite dans une formule introduite par une règle $\flat$ où la machine s'arrête dans l'état $\emptyset$. L'exécution est donc toujours finie dans une quasi forme normale □

### 8.1.3 IAM parallèle

Afin de supprimer les cas où la machine s'arrête par manque d'information sur le poids additif, nous allons introduire une *version parallèle* de la IAM. Les états de cette machine sont des sommes formelles d'états de la machine séquentielle. On note 0 la somme vide et l'état particulier $\emptyset$ est remplacé par 0.

Les seules transitions que l'on modifie sont les suivantes :

$$\begin{array}{lcl} & \& & \\ \Gamma^\uparrow(m,a,w(\not p)) & \to & \Gamma_1^\uparrow(m,a,w.p) + \Gamma_2^\uparrow(m,a,w.\bar{p}) \\ & & \\ & s^\flat & \\ \Gamma^\uparrow(m,a,w(\not p)) & \to & \Gamma_1^\uparrow(m,a,w.p) + \Gamma_2^\uparrow(m,a,w.\bar{p}) \\ \Delta^\uparrow(m,a,w(\not p)) & \to & \Delta_1^\uparrow(m,a,w.p) + \Delta_2^\uparrow(m,a,w.\bar{p}) \end{array}$$

On note $f_\pi^p$ la fonction partielle associée à cette machine et définie comme $f_\pi$ (définition 8.12). Elle associe à une paire (formule conclusion, jeton), une somme formelle de telles paires (éventuellement 0).

8.17. Lemme (Simulation)
*Si $f_\pi^p(A,(m,a,w)) = \sum_i (A_i,(m_i,a_i,w_i))$ alors $f_\pi(A,(m,a,w_i) = (A_i,(m_i,a_i,w_i))$.*

Démonstration : Par récurrence sur le nombre d'états empruntés dans la machine parallèle par le jeton qui arrive à $(A_i,(m_i,a_i,w_i))$, lorsqu'il va de $(A,(m,a,w))$ à $(A_i,(m_i,a_i,w_i))$, en montrant que chaque transition pour ce jeton est réalisée par la machine séquentielle. □

De manière à simplifier les résultats, on considère les sommes formelles modulo :

$$(A,(m,a,w.p)) + (A,(m,a,w.\bar{p})) \equiv (A,(m,a,w))$$

8.18. Proposition (Correction de la machine parallèle)
*Si $\pi$ est une preuve de $\mathsf{MALL}^\flat$ et $\pi_0$ est une quasi forme normale de $\pi$, $f_\pi^p = f_{\pi_0}^p$.*

Démonstration : De la même manière que pour la machine séquentielle en remarquant que les cas pathologiques se résolvent. □

L'effet de la machine parallèle est de calculer dynamiquement l'information de poids nécessaire de manière à ne pas se bloquer inutilement.

### 8.1.4 Constantes

Le problème de l'ajout des constantes est principalement celui de l'interaction entre $\perp$ et &. Si on considère la preuve suivante qui se réduit en l'une des deux preuves possibles de $\vdash 1 \oplus 1$ selon la valeur de $i$ :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash 1}\,1}{\vdash 1 \oplus 1}\,\oplus_1}{\vdash 1 \oplus 1, \perp}\,\perp \quad \dfrac{\dfrac{\dfrac{}{\vdash 1}\,1}{\vdash 1 \oplus 1}\,\oplus_2}{\vdash 1 \oplus 1, \perp}\,\perp}{\vdash 1 \oplus 1, \perp \,\&\, \perp}\,\& \quad \dfrac{\dfrac{}{\vdash 1}\,1}{\vdash 1 \oplus 1}\,\oplus_i}{\vdash 1 \oplus 1}\,cut$$

en terme de réseau, les deux $\perp$ sont déconnectés des preuves de $\vdash 1 \oplus 1$ et il n'y a donc pas de chemin dans la preuve (sans regarder d'éventuelle information de boîte additive ou de poids) qui relie un côté du & à la preuve de $\vdash 1 \oplus 1$ correspondante. Si l'on interprète $\perp$ comme étant déconnecté de son contexte, la GoI ne peut pas distinguer les deux preuves obtenues pour $i = 1$ et $i = 2$. On va donc être amenés comme dans [Gir95a] à rattacher $\perp$ à une formule du contexte ce qui correspond au codage $\perp = \exists\alpha(\alpha \otimes \alpha^\perp)$.

On considère les règles suivantes pour les constantes dans $\mathsf{MALL}^\flat$ :

$$\dfrac{}{\vdash 1}\,1 \qquad \dfrac{\vdash \Gamma_1, A_1}{\vdash \Gamma, A, \perp}\,\perp \qquad \dfrac{\vdash \Gamma_1, \flat}{\vdash \Gamma, \top}\,\top$$

on distingue explicitement une formule du contexte dans la règle $\perp$ afin de résoudre le problème décrit ci-dessus. La règle $\top$ est également modifiée de manière à coller avec l'idée que toute preuve partielle peut être considérée comme complétée par l'introduction de $\top$ et de façon à pouvoir linéariser la réduction.

En ce qui concerne l'élimination des coupures, la réduction multiplicative est la même que dans $\mathsf{MALL}$, et la réduction $\top$ devient une étape commutative :

$$\dfrac{\dfrac{\vdash \Gamma_2, A_1, \flat}{\vdash \Gamma_1, A, \top_1}\,\top \quad \vdash \Delta_1, A^\perp}{\vdash \Gamma, \Delta, \top}\,cut \qquad \rightarrow \qquad \dfrac{\dfrac{\vdash \Gamma_2, A, \flat \quad \vdash \Delta_2, A^\perp}{\vdash \Gamma_1, \Delta_1, \flat}\,cut}{\vdash \Gamma, \Delta, \top}\,\top$$

La traduction $(.)^\natural$ s'étend de manière immédiate à ces nouvelles règles en supprimant la prémisse des règles $\top$ pour retrouver la règle usuelle.

L'alphabet de la pile multiplicative des jetons est étendu et devient $\{g, d, \Uparrow, \Downarrow\}$. Les transitions pour les nouvelles règles sont :

$$\begin{array}{lcllcll}
 & 1 & & & \perp & & \\
1^\uparrow(m,a,w) & \rightarrow & 1^\downarrow(m,a,w) & A^\uparrow(m,a,w) & \rightarrow & \perp^\downarrow(\Uparrow.m,a,w) & \\
 & & & A_1^\downarrow(m,a,w) & \rightarrow & \perp^\downarrow(\Downarrow.m,a,w) & \\
 & & & \perp^\uparrow(\Uparrow.m,a,w) & \rightarrow & A_1^\uparrow(m,a,w) & \\
 & \top & & \perp^\uparrow(\Downarrow.m,a,w) & \rightarrow & A^\downarrow(m,a,w) & \\
\top^\uparrow(m,a,w) & \rightarrow & \top^\downarrow(m,a,w) & \perp^\uparrow(m,a,w) & \rightarrow & \emptyset & m \neq \Uparrow.m', \Downarrow.m' \\
\Gamma^\uparrow(m,a,w) & \rightarrow & \Gamma_1^\uparrow(m,a,w) & \Gamma^\uparrow(m,a,w) & \rightarrow & \Gamma_1^\uparrow(m,a,w) & \\
\Gamma_1^\downarrow(m,a,w) & \rightarrow & \Gamma^\downarrow(m,a,w) & \Gamma_1^\downarrow(m,a,w) & \rightarrow & \Gamma^\downarrow(m,a,w) &
\end{array}$$

8.19. Théorème (Correction avec constantes)
*Le théorème 8.14 est toujours vrai pour* $\mathsf{MALL}^\flat$ *avec constantes.*

Démonstration : On considère le cas d'une coupure $1/\perp$ :

$$\dfrac{\dfrac{\dfrac{\pi_1}{\vdash \Gamma_1, A_1}}{\vdash \Gamma_2, A_2, \perp}\perp \qquad \dfrac{}{\vdash 1}1}{\vdash \Gamma, A}\mathit{cut} \qquad \rightarrow \qquad \dfrac{\pi_1}{\vdash \Gamma, A}$$

La fonction $f_\pi$ est obtenue en composant $f_{\pi_1}$ à gauche et à droite par la transformation $t$ due au passage dans les règles $\perp$ et 1. On doit montrer que ceci ne modifie pas $f_{\pi_1}$. D'une part $t$ ne modifie pas les jetons passant dans $\Gamma$ et d'autre part si un jeton $(m, a, w)$ arrive en montant dans $A$, il suit les étapes $A_2^\uparrow(m, a, w)$, $\perp^\downarrow(\Uparrow .m, a, w)$, $1^\uparrow(\Uparrow .m, a, w)$, $1^\downarrow(\Uparrow .m, a, w)$, $\perp^\uparrow(\Uparrow .m, a, w)$, $A_1^\uparrow(m, a, w)$, il n'a donc pas été modifié, et de même si $(m, a, w)$ arrive en descendant dans $A_1$.

Le cas de la coupure $\top$ ne pose aucun problème. □

On a montré que la IAM fournit un modèle dénotationnel de $\mathsf{MALL}^\flat$, et on va désormais s'intéresser à ce que cela induit pour $\mathsf{MALL}$. On ne peut pas espérer obtenir un modèle correct de $\mathsf{MALL}$ dans le cas général pour toutes les différentes raisons qui nous ont poussés à modifier les règles de $\mathsf{MALL}$ dans $\mathsf{MALL}^\flat$. On cherche donc à montrer un résultat restreint aux preuves de $\vdash 1 \oplus 1$ qui correspond à un codage possible des booléens puisqu'il y a exactement deux preuves distinctes sans coupure de ce séquent.

8.20. Lemme
*Si* $\pi$ *est une preuve de* $\vdash 1$ *dans* $\mathsf{MALL}^\flat$*, il existe un poids* $w$ *tel que* $f_\pi(1, (\varepsilon, \varepsilon, w)) = (1, (\varepsilon, \varepsilon, w))$.

Démonstration : Par le théorème 8.19, on peut se restreindre au cas d'une preuve en quasi forme normale. On raisonne par récurrence sur la taille de cette preuve. Si la dernière règle est une règle 1, le résultat est prouvé avec $w = 1$. Sinon la dernière règle ne peut être qu'une règle $s^\flat$ de poids propre $p$ et dans ce cas on applique l'hypothèse de récurrence à la prémisse $\vdash 1$ ce qui nous donne un poids $w$ et le résultat est prouvé pour le poids $w.p$. □

8.21. Lemme
*Si* $\pi$ *est une preuve de* $\vdash 1, \flat$*, alors pour tout* $e$*,* $f_\pi(e) = \emptyset$.

Démonstration : Par le théorème 8.19, on peut se restreindre au cas d'une preuve en quasi forme normale. On raisonne par récurrence sur la taille de cette preuve. Si la dernière règle est une règle $\flat$, le résultat est immédiat. Si la dernière règle est une coupure, par propriété des quasi formes normales, la formule 1 est introduite par une règle $\flat$ puis n'est contexte que de règles *cut* et le résultat est immédiat. Si la dernière règle est une règle $s^\flat$, on applique l'hypothèse de récurrence à chacune des prémisses. □

8.22. Lemme
*Si* $\pi$ *est une preuve de* $\vdash 1 \oplus 1, \flat$*, alors pour tout* $e$*,* $f_\pi(e) = \emptyset$.

Démonstration : Par le théorème 8.19, on peut se restreindre au cas d'une preuve en quasi forme normale. On raisonne par récurrence sur la taille de cette preuve comme dans le lemme précédent avec comme cas supplémentaire que la dernière règle peut être une règle $\oplus$ et dans ce cas on applique le lemme. □

8.23. Théorème (Résultat booléen)
*Si $\pi$ est une preuve de $\vdash 1 \oplus 1$, et $\pi_0$ une quasi forme normale de $\pi$ alors*

$$\pi_0^\natural = \cfrac{\cfrac{}{\vdash 1}\,1}{\vdash 1 \oplus 1}\,\oplus_i$$

*et*

- *soit il existe $w$ tel que $f_\pi(1 \oplus 1, (\varepsilon, g, w)) = (1 \oplus 1, (\varepsilon, g, w))$ et dans ce cas $i = 1$,*
- *soit il existe $w$ tel que $f_\pi(1 \oplus 1, (\varepsilon, d, w)) = (1 \oplus 1, (\varepsilon, d, w))$ et dans ce cas $i = 2$.*

Démonstration : Le fait que la preuve $\pi_0^\natural$ soit de la forme annoncée est immédiat puisque ce sont les deux seules preuves sans coupure de $\vdash 1 \oplus 1$ dans MALL.

Supposons que $i = 1$, on raisonne par récurrence sur la taille de $\pi_0$.

- Si la dernière règle de $\pi_0$ est une règle $\oplus$, c'est nécessairement une règle $\oplus_1$ par définition de la traduction $(.)^\natural$. On applique le lemme 8.20 à la prémisse ce qui nous donne un poids $w$ tel que $f_{\pi_0}(1\oplus 1, (\varepsilon, g, w)) = (1\oplus 1, (\varepsilon, g, w))$. De plus pour tout poids $w'$, $f_{\pi_0}(1\oplus 1, (\varepsilon, d, w')) = \emptyset$, on ne peut donc pas être dans le deuxième cas.
- Si la dernière règle de $\pi_0$ est une règle $s^\flat$, soit $p$ son poids propre. On applique l'hypothèse de récurrence à la prémisse $\vdash 1 \oplus 1$ et on obtient un poids $w$ tel que $f_{\pi_0}(1 \oplus 1, (\varepsilon, g, w.p)) = (1 \oplus 1, (\varepsilon, g, w.p))$. De plus pour tout poids $w'$, $f_{\pi_0}(1 \oplus 1, (\varepsilon, d, w')) = \emptyset$ par hypothèse de récurrence et par le lemme 8.22.

On conclut finalement par le théorème 8.19. □

8.24. Remarque : Il n'est pas possible de se restreindre au cas $w = 1$ dans ce théorème puisqu'il existe des preuves $\pi$ pour lesquelles $f_\pi(A, (m, a, 1)) = \emptyset$ pour toute conclusion $A$ et tout jeton $(m, a, 1)$ de poids 1 (par exemple la preuve de $\vdash 1 \oplus 1$ page 125).

À nouveau la machine parallèle permet de résoudre ce défaut puisque le poids peut être construit dynamiquement.

8.25. Proposition (Résultat booléen parallèle)
*Si $\pi$ est une preuve de $\vdash 1 \oplus 1$, et $\pi_0$ une quasi forme normale de $\pi$ alors*

$$\pi_0^\natural = \cfrac{\cfrac{}{\vdash 1}\,1}{\vdash 1 \oplus 1}\,\oplus_i$$

*et*

- *soit $f^p_\pi(1 \oplus 1, (\varepsilon, g, 1)) \neq 0$ et dans ce cas $i = 1$,*
- *soit $f^p_\pi(1 \oplus 1, (\varepsilon, d, 1)) \neq 0$ et dans ce cas $i = 2$.*

Démonstration : Par la proposition 8.18, il suffit de se restreindre au cas où $\pi = \pi_0$. On suppose que $i = 1$ et on raisonne par récurrence sur la taille de $\pi_0$. Si la dernière règle est une règle $\oplus_i$ appliquée à une preuve $\pi'$ de $\vdash 1$, on a facilement $f^p_{\pi'}(1, (\varepsilon, \varepsilon, 1)) \neq 0$ ce qui permet de conclure que $f^p_{\pi_0}(1 \oplus 1, (\varepsilon, g, 1)) \neq 0$ et $f^p_{\pi_0}(1 \oplus 1, (\varepsilon, d, 1)) = 0$. Sinon la dernière règle est une règle $s^\flat$ et on applique l'hypothèse de récurrence à la sous-preuve $\pi'$ de $\vdash 1 \oplus 1$ ce qui nous donne $f^p_{\pi'}(1 \oplus 1, (\varepsilon, g, 1)) \neq 0$ et $f^p_{\pi'}(1 \oplus 1, (\varepsilon, d, 1)) = 0$. On en déduit d'une part que $f^p_{\pi_0}(1 \oplus 1, (\varepsilon, g, 1)) \neq 0$ et d'autre part que $f^p_{\pi_0}(1 \oplus 1, (\varepsilon, d, 1)) = 0$, sinon la valeur non nulle proviendrait nécessairement de la prémisse $\vdash 1 \oplus 1, \flat$ ce qui est impossible par les lemmes 8.17 et 8.22. □

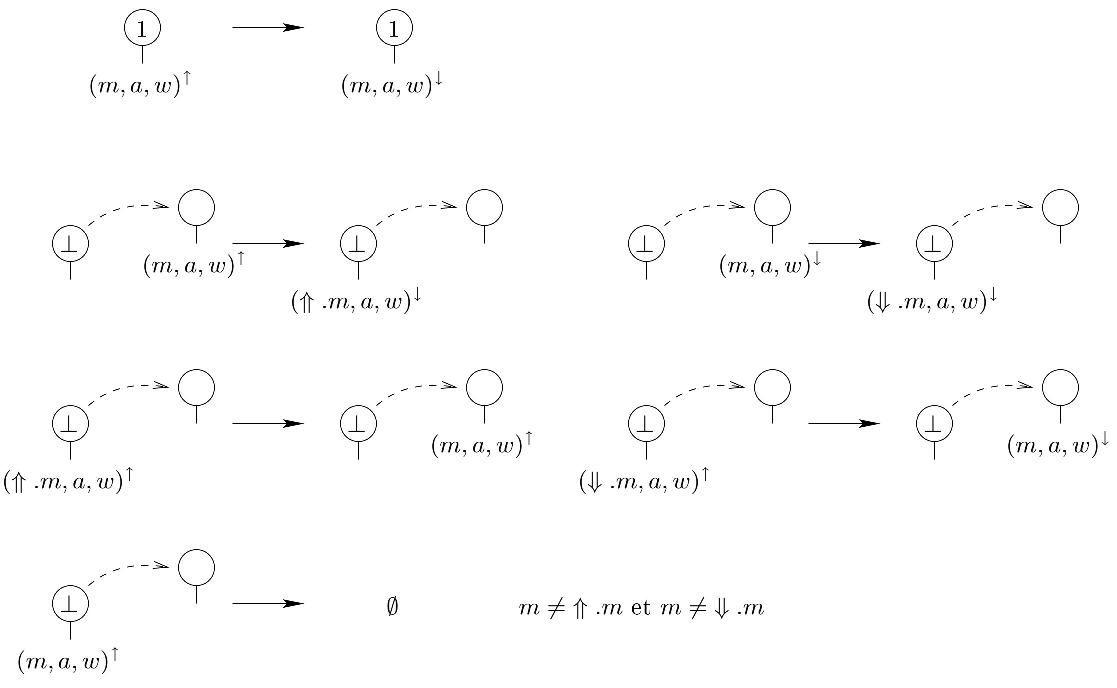
### 8.1.5 Interprétation des réseaux

Nous avons interprété les preuves du calcul des séquents mais sans réellement utiliser la séquentialité des règles. Il paraît naturel de vouloir définir la IAM directement sur les réseaux comme c'est le cas pour MELL. La difficulté vient du fait qu'il n'y a pas de notion de réseaux pour $\mathsf{MALL}^\flat$ avec élimination des coupures qui permette de faire entièrement dans les réseaux ce que l'on a fait en utilisant le calcul des séquents. Nous allons donc devoir nous contenter de l'approche suivante : définir la IAM sur les réseaux, montrer que l'interprétation d'une preuve du calcul des séquents ne dépend en fait que du réseau associé et puisque la forme normale d'une preuve du calcul des séquents est une séquentialisation de la forme normale du réseau associé, on en conclura le résultat de correction pour les réseaux de conclusion $1 \oplus 1$.

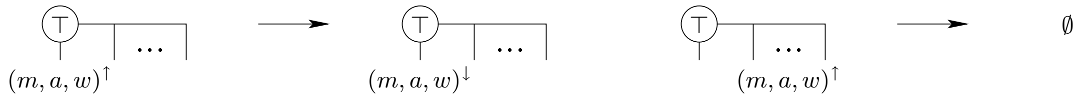
Ce résultat ne pourrait pas être obtenu avec l'approche de Girard dans [Gir95a] car celle-ci dépend trop de la séquentialité des preuves.

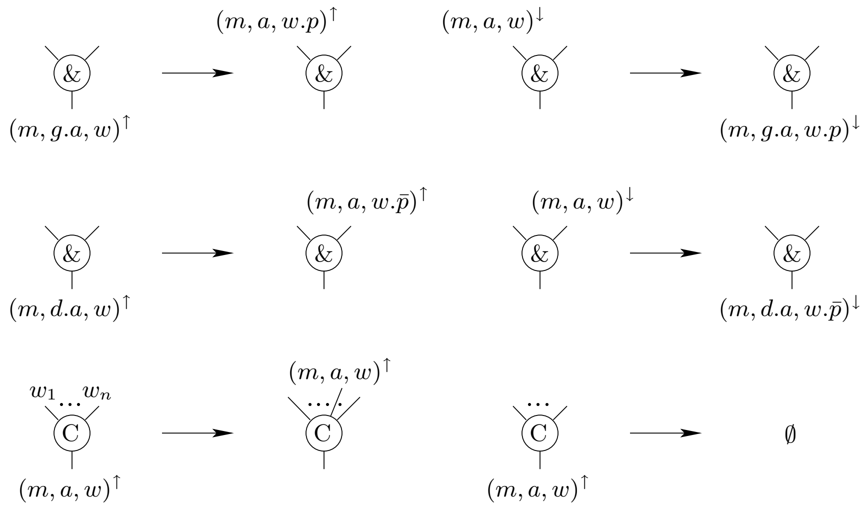
On utilise ici les réseaux à poids pour MALL tels qu'ils ont été décrits par Girard dans [Gir96]. Il s'agit essentiellement de la même définition que pour les réseaux polarisés à poids (section 5.2) excepté que d'une part les poids doivent être des monômes et d'autre part afin de résoudre le problème donné dans l'exemple 1.12 pour la correction des réseaux en présence de $\perp$ et celui donné au début de la section 8.1.4, à chaque nœud $\perp$ d'une structure de preuve est associé un nœud non *cut* appelé *saut* du nœud $\perp$.

On définit la IAM comme pour le cas de MELL avec les mêmes transitions pour les nœuds *ax*, *cut*, $\otimes$ et $\parr$. Pour les autres nœuds :

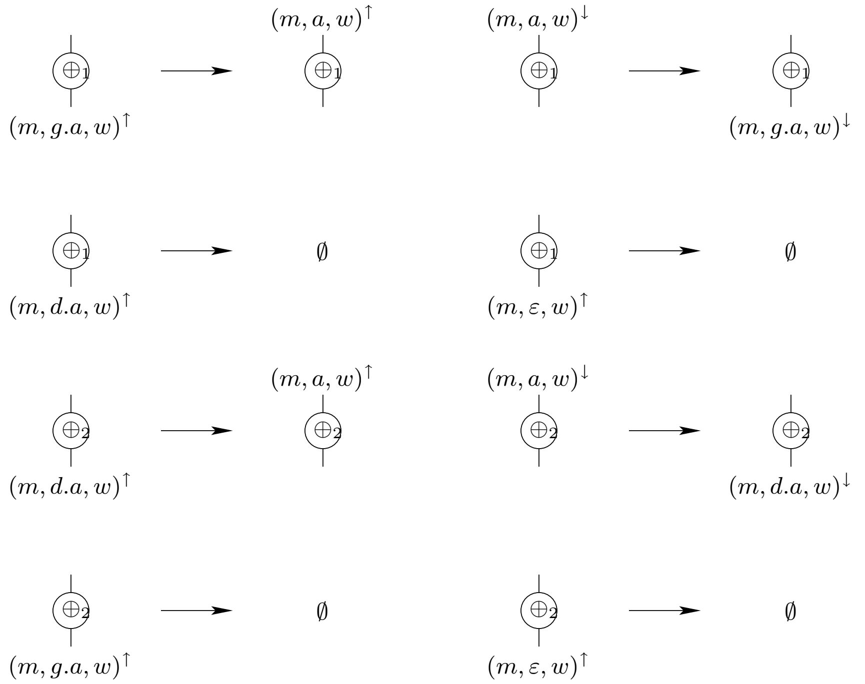



la flèche indique le saut du nœud $\perp$, qui est une information utilisée explicitement par la machine alors qu'il ne s'agissait a priori que d'un outil pour la correction.






si le poids $w_k$ de la prémisse est tel que $w \leq w_k$ sinon

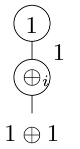



La fonction d'interaction $f_{\mathcal{R}}$ est obtenue comme précédemment.

8.26. PROPOSITION
*Soit $\mathcal{R}$ un réseau, si $\mathcal{R}$ se séquentialise en $\pi$ alors $f_{\mathcal{R}} = f_{\pi}$.*

Démonstration : Par récurrence sur la taille de $\mathcal{R}$, on remarquant que les contextes de règles ne modifient pas le jeton dans les transitions de la machine pour le calcul des séquents. Le seul cas plus complexe est celui d'une règle & mais les opérations sur le contexte correspondent exactement aux transitions des nœuds C ce qui donne le résultat voulu. □

8.27. COROLLAIRE (RÉSULTAT BOOLÉEN)
*Si $\mathcal{R}$ est un réseau de conclusion $1 \oplus 1$, et $\mathcal{R}_0$ sa forme normale, alors $\mathcal{R}_0$ est :*



*et*

- *soit il existe $w$ tel que $f_{\mathcal{R}}(1 \oplus 1, (\varepsilon, g, w)) = (1 \oplus 1, (\varepsilon, g, w))$ et dans ce cas $i = 1$,*
- *soit il existe $w$ tel que $f_{\mathcal{R}}(1 \oplus 1, (\varepsilon, d, w)) = (1 \oplus 1, (\varepsilon, d, w))$ et dans ce cas $i = 2$.*

Démonstration : Soit $\pi$ une séquentialisation de $\mathcal{R}$ et $\pi_0$ sa forme normale (dans MALL), d'après la proposition précédente, on a $f_{\mathcal{R}} = f_\pi$ et $f_{\mathcal{R}_0} = f_{\pi_0}$. On conclut alors par le théorème 8.23. □

8.28. Remarque : On pourrait également définir la machine parallèle associée à un réseau et étendre aux réseaux les résultats du calcul des séquents.

## 8.2 GoI parallèle pour MELL

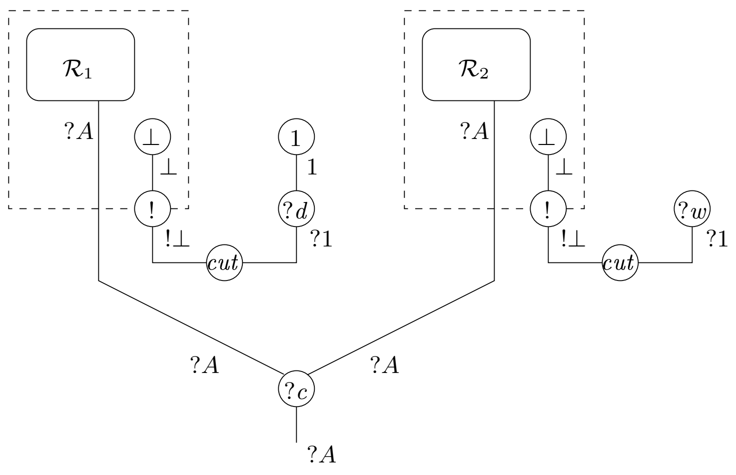
On considère ici le système MELL avec promotion fonctorielle décrit dans la remarque 2.15.

La difficulté du passage de la GoI pour LL à la GoI pour LLP réside dans les règles structurelles généralisées. Les problèmes à résoudre sont de nature proche de ceux rencontrés dans la gestion des contractions additives (que doit-on faire si toutes les conclusions d'une preuve sont conclusions de contractions ?). Nous allons montrer que l'approche par *machine à jetons parallèle* s'étend à LLP, en revenant tout d'abord sur le cas de MELL.

Dans le cas de MELL, la correction de la GoI est prouvée avec des contraintes sur les preuves et la réduction (remarque 2.15), en particulier on ne considère que les réductions $!/?w$ avec des boîtes sans porte auxiliaire. Autrement dit cela signifie que l'on n'efface aucun chemin entrant dans une boîte par une porte auxiliaire et ressortant également par une porte auxiliaire.

Dans le cas particulier des réseaux qui sont des traductions de $\lambda$-termes, cette contrainte ne porte pas à conséquence, malheureusement lorsque l'on souhaite étendre ceci au $\lambda\mu$-calcul, la contrainte devient trop forte. Sans vouloir entrer dans les détails de la traduction du $\lambda\mu$-calcul dans LLP qui sera décrite dans la partie suivante, si l'on veut raisonnablement appliquer des résultats de GoI au $\lambda\mu$-calcul via cette traduction, il est nécessaire de pouvoir libéraliser la contrainte utilisée pour MELL.

Le seul moyen de distinguer la preuve suivante de celle où $\mathcal{R}_1$ et $\mathcal{R}_2$ sont inversés est de pouvoir constater que la porte principale de la boîte contenant $\mathcal{R}_2$ est coupée sur un affaiblissement, or ceci est impossible avec la machine à jetons de MELL.



Nous allons décrire la machine parallèle pour MELL directement dans les réseaux avec l'idée que les bords de boîtes ! servent de points de synchronisation. Ceci va nous permettre de prendre correctement en compte les chemins qui entrent et sortent d'une boîte par ses portes auxiliaires.

On modifie légèrement le notion de jeton donnée à la définition 2.8 :

8.29. Définition (Jeton)
Un *jeton* est un triplet $(m, b, s)$ où :
- $m$ est un jeton multiplicatif,
- $b$ et $s$ sont des piles exponentielles, où une signature exponentielle est donnée par :

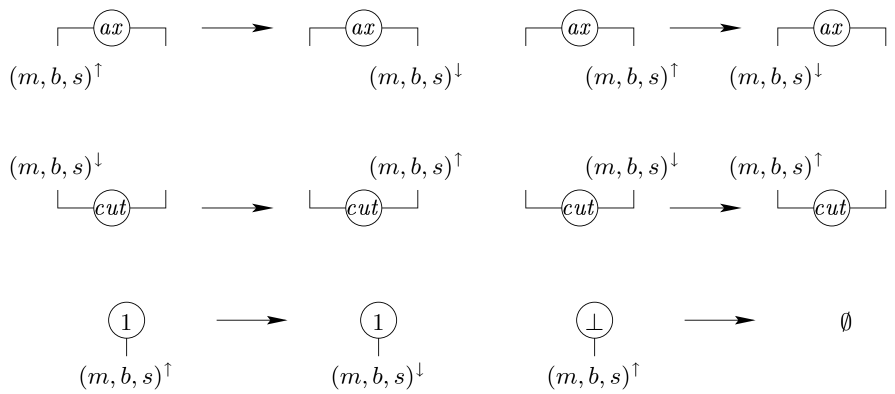
$$\sigma \quad ::= \quad \circ \quad | \quad \star \quad | \quad g.\sigma \quad | \quad d.\sigma \quad | \quad \ulcorner\sigma\urcorner.\sigma$$

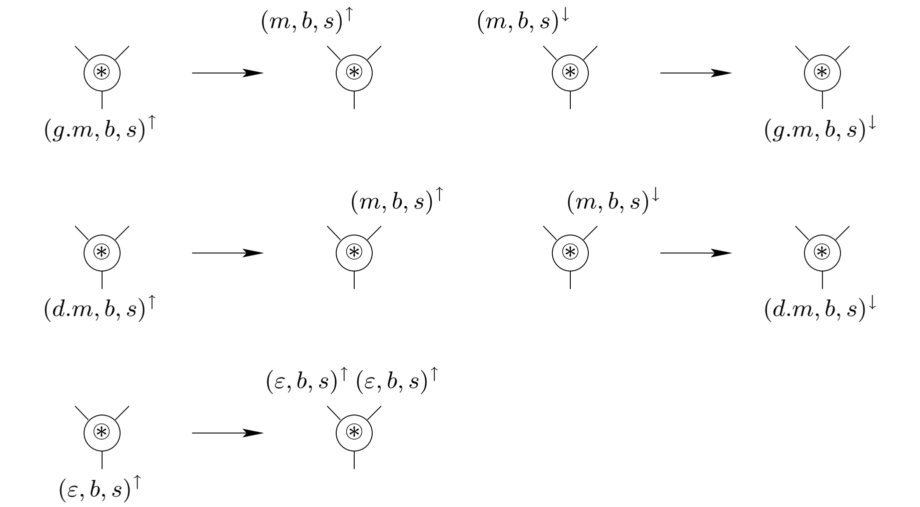
8.30. Définition (Substitution dans une signature exponentielle)
Soit $\sigma$ une signature exponentielle, la signature exponentielle $\hat{\sigma}$ est une *substitution* de $\sigma$ si elle est obtenue en remplaçant des occurrences de $\circ$ dans $\sigma$ par des signatures exponentielles ou, plus formellement :
- toute signature exponentielle est une substitution de $\circ$,
- seule $\star$ est une substitution de $\star$,
- si $\hat{\sigma}$ est une substitution de $\sigma$ alors $g.\hat{\sigma}$ est une substitution de $g.\sigma$ et $d.\hat{\sigma}$ est une substitution de $d.\sigma$,
- si $\hat{\sigma}_1$ est une substitution de $\sigma_1$ et $\hat{\sigma}_2$ est une substitution de $\sigma_2$ alors $\ulcorner\hat{\sigma}_1\urcorner.\hat{\sigma}_2$ est une substitution de $\ulcorner\sigma_1\urcorner.\sigma_2$.

On étend cette notion aux piles exponentielles : une substitution d'une pile exponentielle est obtenue en remplaçant les éléments de la pile par des substitutions.

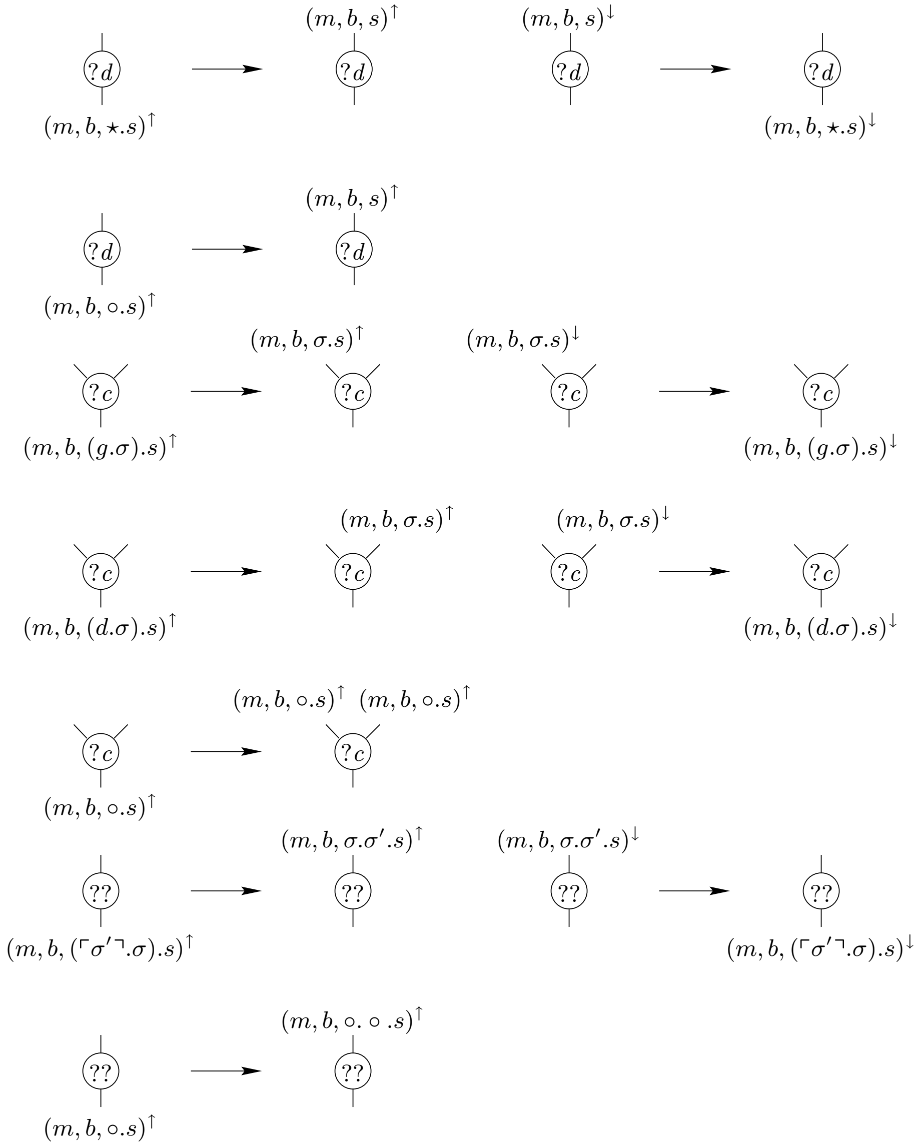
8.31. Définition (Machine abstraite parallèle)
Soit $\mathcal{R}$ un réseau de MELL, on lui associe la machine $M_{\mathcal{R}}$ dont un état est donné par :
- pour chaque arête, un ensemble de jetons valides avec chacun une direction de déplacement $\uparrow$ ou $\downarrow$ ;
- pour chaque nœud !, un ensemble de piles exponentielles.

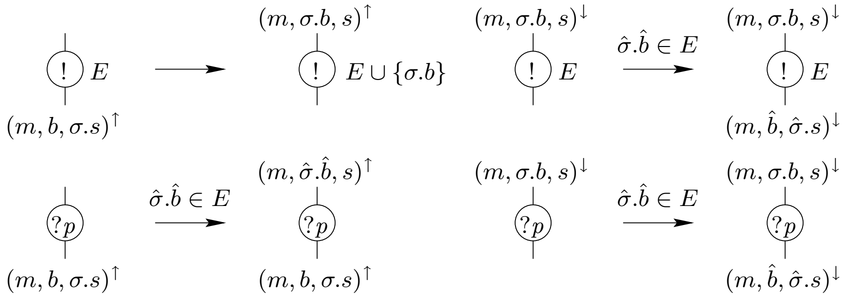
Les transitions de la machine sont définies pour chaque nœud du réseau et concernent un jeton placé sur l'une des arêtes de ce nœud. Si $\emptyset$ est le résultat d'une transition, cela signifie que le jeton est effacé. On note $e \to e'$ le fait qu'il existe une transition de la machine permettant de passer de l'état $e$ à l'état $e'$.






où $\circledast$ est soit un nœud $\otimes$ soit un nœud $⅋$.






où $E$ est l'ensemble associé au nœud ! correspondant.

L'*état initial* de la machine associe un ensemble vide de piles aux nœuds ! et un jeton $(\varepsilon, \varepsilon, \circ^n)$ avec la direction $\uparrow$ dans chaque conclusion du réseau (où $n$ est le nombre d'exponentielles en tête dans le type de la conclusion).

Un état $e$ est *accessible* s'il existe une suite de transitions qui mène de l'état initial à $e$.

8.32. Remarque : La validité des jetons est bien préservée par toutes les transitions de la machine.

Si la pile $b$ d'un jeton est non vide (ce qui est équivalent à dire que le jeton n'est pas à profondeur 0), elle ne peut pas être modifiée si le jeton ne sort pas de la boîte dans laquelle il se trouve.

8.33. Lemme (Confluence)
*Soit $\mathcal{R}$ un réseau et $e_1$ un état de $M_{\mathcal{R}}$, si $e_1 \to e_2$ et $e_1 \to e_3$ alors il existe un état $e_4$ tel que $e_2 \to e_4$ et $e_3 \to e_4$.*

Démonstration : Il suffit de remarquer que l'application de la transition qui mène de $e_1$ à $e_2$ ne peut pas empêcher d'appliquer ensuite celle qui aurait mené à $e_3$ puisque l'application d'une transition ne peut qu'augmenter l'espace des transitions possibles (dans le cas d'un jeton arrivant en montant dans un nœud ! par exemple). $\square$

8.34. Proposition (Simulation)
*Soient $\mathcal{R}$ un réseau, $\mathcal{R}'$ un réduit de $\mathcal{R}$ et $j$ un jeton, s'il existe un état accessible de $M_{\mathcal{R}}$ contenant le jeton $j$ descendant dans une conclusion $A$ alors il existe un état accessible de $M_{\mathcal{R}'}$ contenant le jeton $j$ descendant dans la même conclusion $A$.*

Démonstration : On suppose que $\mathcal{R}'$ est obtenu à partir de $\mathcal{R}$ par une étape de réduction, sinon il suffit de raisonner par récurrence sur le nombre d'étapes entre $\mathcal{R}$ et $\mathcal{R}'$.

On raisonne de la façon suivante : on définit une fonction $f$ qui, à chaque état accessible de $M_{\mathcal{R}}$, associe un état de $M_{\mathcal{R}'}$ de manière à ce que :

1. $e'_i \to f(e_i)$ où $e_i$ est l'état initial de $M_{\mathcal{R}}$ et $e'_i$ celui de $M_{\mathcal{R}'}$,
2. si $e_1 \to e_2$ alors $f(e_1) \to^* f(e_2)$,
3. si, dans l'état $e$, le jeton $j$ descend dans une conclusion $A$ de $\mathcal{R}$ alors c'est également le cas dans $f(e)$,

On considère chaque cas d'élimination des coupures en utilisant les notations de la section 1.3, modifiées à la fin de la section 2.2 :

($ax$) On définit $f$ en associant à tout jeton dans la prémisse de type $A$ de la coupure ou dans la conclusion de type $A$ non coupée de l'axiome, le même jeton dans l'arête de type $A$ obtenue et à tout jeton montant (resp. descendant) dans l'arête de type $A^{\perp}$, le même jeton descendant (resp. montant) dans l'arête de type $A$. Les autre jetons sont juste recopiés par $f$ dans les arêtes correspondantes et les ensembles associés aux nœuds ! ne sont pas modifiés.

1. L'état initial est préservé par $f$ puisque, si la conclusion $A$ de l'axiome est une conclusion de $\mathcal{R}$, le jeton qu'elle contient n'est pas modifié.
2. $f$ ne fait qu'identifier des états qui s'obtenaient par réduction.
3. Immédiat.

($\otimes/\parr$) On définit $f$ en remplaçant tout jeton $(g.m, b, s)^{\uparrow}$ (resp. $(g.m, b, s)^{\downarrow}$) dans une prémisse de la coupure par le jeton $(m, b, s)^{\uparrow}$ (resp. $(m, b, s)^{\downarrow}$) dans la prémisse correspondante de la coupure entre $A$ et $A^{\perp}$ (et de même avec $d$ dans la coupure entre $B$ et $B^{\perp}$).

On remarque que $f$ est bien définie sur tout état accessible de $M_{\mathcal{R}}$ puisqu'un tel état ne peut pas contenir de jeton $(\varepsilon, b, s)^{\updownarrow}$ dans une prémisse de la coupure.

1. Immédiat, tout comme 3.
2. $f$ ne fait qu'identifier des états qui s'obtenaient par réduction.

($1/\perp$) $f$ est tout simplement l'identité pour les arêtes qui ne sont pas les prémisses de la coupure et oublie les jetons présents dans les prémisses de la coupure. Les différentes propriétés sont immédiates.

($!/?d$) Soit $e$ un état accessible de $M_{\mathcal{R}}$ et $E$ l'ensemble de piles associé au nœud ! (qui ne peut contenir que des éléments de la forme $\star.b$), s'il y a un jeton dans la boîte alors il est de la forme $(m, \star.b, s)$, en effet il est de la forme $(m, \sigma.b, s)$ par validité et :

- soit il est entré par la porte principale et il est nécessairement passé par le nœud $?d$ donc $\sigma = \star$,
- soit il est entré par une porte auxiliaire et $\sigma.b \in E$ donc $\sigma = \star$,

à un tel jeton, on associe par $f$ le jeton $(m, b, s)$ dans l'arête correspondante avec la même direction. Si $e$ contient un jeton dans la prémisse $!A$ (resp. $?A^{\perp}$) de la coupure alors, comme ci-dessus, il est de la forme $(m, \star.b, s)$ et on lui associe le jeton $(m, b, s)$ dans la prémisse $A$ (resp. $A^{\perp}$) de la coupure dans $\mathcal{R}'$. Les autre jetons ne sont pas modifiés par $f$.

1. Immédiat, tout comme 3.
2. La seule transition à vérifier est celle qui concerne les portes auxiliaires de la boîte qui est bien simulée par le nœud $?d$. Les autres sont immédiates.

($!/?w$) $f$ supprime tout simplement les jetons contenus dans les arêtes effacées par la réduction. Les propriétés sont immédiates.

($!/?c$) Soit $e$ un état accessible de $M_{\mathcal{R}}$ et $E$ l'ensemble de piles associé au nœud !, $E$ est constitué de piles $(g.\sigma).b$ et $(d.\sigma).b$, on note $E_g = \{\sigma.b \mid (g.\sigma).b \in E\}$ et $E_d = \{\sigma.b \mid (d.\sigma).b \in E\}$. On associe $E_g$ à la boîte coupée sur la prémisse gauche de la contraction et $E_d$ à celle coupée sur la prémisse droite.

- à un jeton $(m, b, (g.\sigma).s)$ (resp. $(m, b, (d.\sigma).s)$) dans une prémisse de la coupure, on associe le jeton $(m, b, \sigma.s)$ dans la coupure gauche (resp. droite),
- à un jeton $(m, (g.\sigma).b, s)$ (resp. $(m, (d.\sigma).b, s)$) dans la boîte, on associe le jeton correspondant $(m, \sigma.b, s)$ dans la boîte gauche (resp. droite),
- à un jeton $(m, b, \circ.s)$ montant dans la conclusion d'une porte auxiliaire de la boîte, on associe un jeton $(m, b, \circ.s)$ dans chaque prémisse de la contraction correspondante,
- à un jeton $(m, b, (g.\sigma).s)$ (resp. $(m, b, (d.\sigma).s)$) montant dans la conclusion d'une porte auxiliaire de la boîte, on associe un jeton $(m, b, \sigma.s)$ dans la prémisse gauche (resp. droite) de la contraction correspondante,
- aux autres jetons dans la conclusion d'une porte auxiliaire de la boîte, on associe le même jeton dans la conclusion de la contraction correspondante.

En ce qui concerne les propriétés :

1. Immédiat, tout comme 3.
2. On vérifie que les transitions sur les portes auxiliaires sont bien simulées, les autres ne posent pas de problèmes. Le point le plus délicat concerne le cas d'un jeton $(m, b, \circ.s)$ qui traverse vers le haut une porte auxiliaire. En effet lorsque l'on simule cette transition dans la machine $M_{\mathcal{R}'}$, le jeton va d'abord monter dans la contraction puis franchir l'une des portes auxiliaires (ce qui correspond à la transition dans $M_{\mathcal{R}}$), le problème est qu'il peut rester bloqué dans l'autre porte auxiliaire et pour que la simulation soit vraie, il faut que l'état obtenu soit l'image de l'état résultant de la transition dans $M_{\mathcal{R}}$, c'est pour cela que la définition de $f$ fait monter automatiquement les jetons $(m, b, \circ.s)$. On a le même type de phénomène avec les jetons $(m, b, (g.\sigma).s)$ et $(m, b, (d.\sigma).s)$ qui montent dans les conclusions de portes auxiliaires de la boîte.

(!/??) Soit $e$ un état accessible de $M_{\mathcal{R}}$ et $E$ l'ensemble de piles associé au nœud !, $E$ est un ensemble de piles de la forme $(\ulcorner\sigma_i'\urcorner.\sigma_i).b_i$, on associe à la boîte extérieure l'ensemble des $\sigma_i.b_i$ et à la boîte intérieure l'ensemble des $\sigma_i'.\sigma_i.b_i$. Concernant les jetons :

- à un jeton dans une prémisse de la coupure donc de la forme $(m, b, (\ulcorner\sigma_i'\urcorner.\sigma_i).s)$, on associe le jeton $(m, b, \sigma_i.\sigma_i'.s)$ dans l'arête correspondante,
- à un jeton $(m, (\ulcorner\sigma_i'\urcorner.\sigma_i).b, s)$ dans la boîte, on associe le jeton $(m, \sigma_i'.\sigma_i.b, s)$ dans l'arête correspondante,
- à un jeton $(m, b, (\ulcorner\sigma_i'\urcorner.\sigma_i).s)$ montant dans une porte auxiliaire de la boîte, on associe le jeton $(m, b, \sigma_i.\sigma_i'.s)$ dans la conclusion de la porte auxiliaire correspondante de la boîte extérieure et le jeton ,$(m, \sigma_i.b, \sigma_i'.s)$ dans celle de la boîte intérieure.

En ce qui concerne les propriétés :

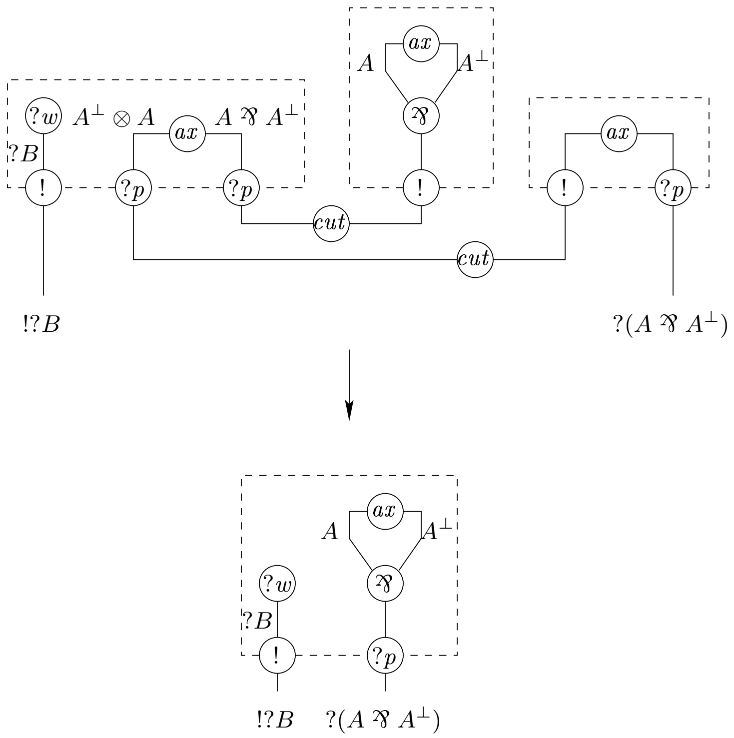
1. Immédiat, tout comme 3.
2. On vérifie que les transitions sur les portes auxiliaires de la boîte sont bien simulées, les autres ne posant pas de problèmes. Le point le plus délicat est celui d'un jeton $(m, b, (\ulcorner\sigma_i'\urcorner.\sigma_i).s)$ montant dans une porte auxiliaire de la boîte puisqu'après la transition il reste présent dans cette porte auxiliaire dans $M_{\mathcal{R}}$ et dans les portes auxiliaires des deux boîtes dans $M_{\mathcal{R}'}$.

(!/!) Soit $e$ un état accessible de $M_{\mathcal{R}}$, $E_1$ l'ensemble de piles associé au nœud ! de conclusion $!A$, et $E_2$ l'ensemble de piles associé au nœud ! de conclusion $!B$, on a $E_2 \subset E_1$. On associe $E_1$ à l'unique boîte obtenue dans $\mathcal{R}'$ et à chaque jeton $(m, b, \sigma.s)$ dans une prémisse de la coupure, on associe le jeton $(m, \sigma.b, s)$ dans la prémisse correspondante de la coupure dans $\mathcal{R}'$.

1. Immédiat, tout comme 3.
2. Pas de difficultés puisque le fait de remplacer $E_2$ par $E_1$ pour les portes auxiliaires $?\Delta$ autorise plus de transitions dans $M_{\mathcal{R}'}$.

En utilisant $f$ et ses trois propriétés, on peut désormais terminer la preuve : si $e$ est accessible alors $e_i \to^* e$ par définition, donc on a $f(e_i) \to^* f(e)$ par (2) et $e_i' \to f(e_i)$ par (1) donc $f(e)$ est un état accessible de $M_{\mathcal{R}'}$ et par (3), si $j$ descend dans la conclusion $A$ dans $e$, c'est également le cas dans $f(e)$. □

8.35. Exemple
La réciproque de la proposition n'est pas vérifiée, il peut y avoir plus de jetons qui arrivent dans les conclusions de $\mathcal{R}'$ que dans celles de $\mathcal{R}$, si $\mathcal{R} \to^+ \mathcal{R}'$ :




aucun jeton ne peut arriver dans la conclusion $?(A \parr A^\perp)$ de $\mathcal{R}$ alors que les jetons $(g, \varepsilon, \circ)$ et $(d, \varepsilon, \circ)$ arrivent dans celle de $\mathcal{R}'$.

## 8.3 GoI polarisée

### 8.3.1 GoI pour MELLP

De manière à obtenir la réciproque de la proposition 8.34, on va se placer dans un cadre polarisé. On considère les réseaux pour MELLP avec promotion fonctorielle, *i.e.* avec des nœuds $?p$ et ??, ... ce qui impose d'avoir des nœuds $?d$ dont la prémisse a type $N$, et des nœuds ! dont la prémisse a type $P$.

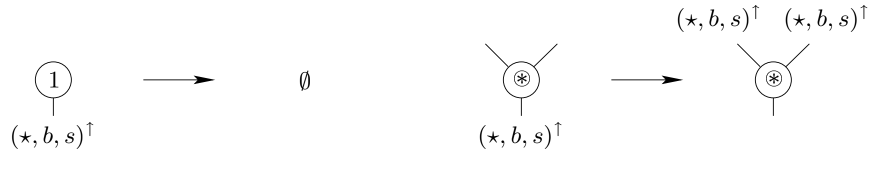
La notion de jeton multiplicatif est étendue avec une constante $\star$ :

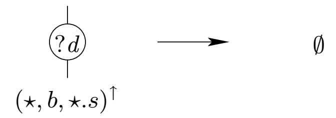
$$m \quad ::= \quad \varepsilon \quad | \quad \star \quad | \quad g.m \quad | \quad d.m$$

Les formules polarisées ayant une structure exponentielle implicite, le type d'un jeton valide doit avoir une signature de plus qu'auparavant dans la pile $s$.

L'*état initial* de la machine associe un ensemble vide de piles aux nœuds !, un jeton $(\star, \circ^k, \circ)$ avec la direction $\downarrow$ dans la conclusion de chaque nœud 1 et ! à profondeur $k$ (où $\circ^k$ est la pile exponentielle contenant $k$ signatures exponentielles qui sont toutes égales à $\circ$), et un jeton $(\varepsilon, \varepsilon, \circ^{n+1})$ avec la direction $\uparrow$ dans chaque conclusion du réseau (où $n$ est le nombre d'exponentielles en tête dans le type de la conclusion).

Les transitions sont les mêmes que pour MELL en ajoutant les transitions suivantes :






La substitution lors de la traversée du bord d'une boîte est remplacée par l'unification.

8.36. Lemme (Accessibilité)
*Soit $\mathcal{R}$ un réseau et $a$ une arête de $\mathcal{R}$ à profondeur $0$, il existe un état accessible de $M_{\mathcal{R}}$ contenant un jeton $(m, \varepsilon, \circ^n)$ dans l'arête $a$ avec une direction opposée à celle de $a$ pour la p-orientation.*

Démonstration : D'après le critère de correction, il existe un chemin du graphe de correction à profondeur 0 qui mène de $a$ à une arête finale, c'est-à-dire une conclusion du réseau ou une conclusion de nœud 1. On raisonne par récurrence sur la longueur du plus court chemin de $a$ à une arête finale.

Si $a$ est une arête finale, l'état initial vérifie la propriété. Sinon, on raisonne selon le nœud $n$ dans lequel $a$ est entrante :

– si $n$ est un nœud *ax*, *cut*, $⅋$, $\otimes$, $?c$ ou $??$, le résultat est immédiat par hypothèse de récurrence ;
– si $n$ est un nœud !, l'état initial vérifie la propriété. □

On utilise la variante de MELLP avec promotion fonctorielle pour obtenir une meilleure adéquation de l'élimination des coupures avec la GoI. Cependant la syntaxe qui nous intéresse réellement est celle de MELLP, on va donc se restreindre pour la proposition suivante au cas d'un réseau qui est la traduction d'un réseau de MELLP.

8.37. Proposition (Simulation)
*Soient $\mathcal{R}$ un réseau dans* MELLP *avec promotion fonctorielle qui est la traduction d'un réseau de* MELLP*, $\mathcal{R}_0$ sa forme normale et $j$ un jeton ; il existe un état accessible de $M_{\mathcal{R}}$ contenant le jeton $j$ descendant dans une conclusion $A$* ssi *il existe un état accessible de $M_{\mathcal{R}_0}$ contenant le jeton $j$ descendant dans la même conclusion $A$.*

Démonstration : On raisonne comme dans la proposition 8.34, ce qui donne la première implication dans tous les cas et la seconde excepté pour les coupures $!/?d$ et $!/!$. L'hypothèse que $\mathcal{R}$ est la traduction d'un réseau de MELLP, nous permet de considérer la stratégie de réduction qui ne fait de réduction $!/!$ que juste après la réduction $!/??$ correspondante. De plus on va appliquer cette stratégie profondeur par profondeur en commençant par la profondeur 0.

On considère une étape de réduction de $\mathcal{R}$ à $\mathcal{R}'$ :

($!/?d$) Si la coupure éliminée est à profondeur 0, par le lemme d'accessibilité, il existe un état accessible $e_0$ de $\mathcal{R}$ tel que qu'il y ait un jeton $(m, \varepsilon, \circ^n)$ descendant dans la prémisse de la déréliction. Si la coupure éliminée est à profondeur $k$, puisqu'il n'y a plus de coupures à profondeur inférieure, il existe un état dans lequel toutes les conclusions de la boîte dans laquelle se trouve la coupure contiennent un jeton montant de la forme $(m, \circ^k, \circ^p)$ (il suffit de faire monter jusque là les jetons placés dans les conclusions du réseau dans l'état initial), on peut alors appliquer le lemme d'accessibilité dans la boîte et on obtient un état accessible $e_0$ tel que qu'il y ait un jeton $(m, \circ^k, \circ^n)$ descendant dans la prémisse de la déréliction.

En appliquant à $e_0$ les transitions correspondant au nœud $?d$, à la coupure et au nœud !, on obtient un état $e_1$ dans lequel l'ensemble $E$ associé au nœud ! contient la pile $\star.\circ^k$. En utilisant une fonction inverse de celle décrite dans la preuve de la proposition 8.34, on voit que la simulation des transitions dans la coupure ne pose pas de problèmes et concernant la simulation des transitions dans les dérélictions dans $\mathcal{R}'$ : si une telle transition a lieu entre les états $e$ et $e'$ alors cette transition est simulée dans $\mathcal{R}$ après avoir appliqué les transitions nécessaires pour passer de $f^{-1}(e)$ à un état dans lequel $E$ contient $\star.\circ^k$, ce qui est possible par confluence.

(!/!) On utilise le fait que l'étape qui précède est une étape !/?? ce qui entraîne que la boîte de droite contient une boîte. À chaque état accessible de $\mathcal{R}'$ on associe un état accessible de $\mathcal{R}$ par :

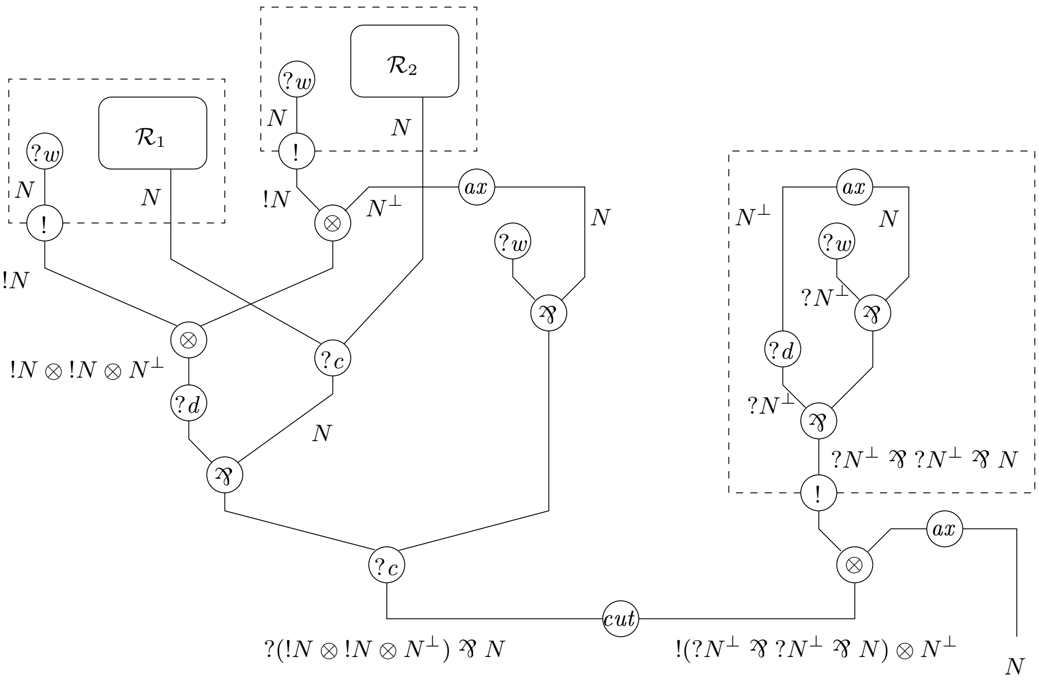
- Si $E_1$ est l'ensemble de piles associé à la boîte de extérieure dans $\mathcal{R}'$, et $E_2''$ celui associé à la boîte intérieure, pour tout $\sigma.b \in E_2''$ on a $b \in E_1$. Dans $\mathcal{R}$, on associe $E_1$ à la boîte de gauche, $E_2' = \{b \mid \sigma.b \in E_2''\}$ à la boîte de droite extérieure et $E_2''$ à la boîte de droite intérieure. On a $E_2' \subset E_1$, en raisonnant comme pour la proposition 8.34.
- À un jeton dans une prémisse de la coupure donc de la forme $(m, \sigma.b, s)$, on associe le jeton $(m, b, \sigma.s)$ dans l'arête correspondante,
- À un jeton dans la boîte extérieure, on associe le même jeton dans l'arête correspondante,
- À un jeton $(m, b, \sigma_1.\sigma_2.s)$ montant dans une porte auxiliaire de la boîte extérieure dont la prémisse est porte auxiliaire de la boîte intérieure, on associe le jeton $(m, b, \sigma_1.\sigma_2.s)$ l'arête correspondante et le jeton $(m, \sigma_1'.b', \sigma_2.s)$ dans l'arête située au-dessus si cela correspond à une transition correcte par rapport à $E_1$ dans $\mathcal{R}'$.

On vérifie alors que les transitions de $\mathcal{R}'$ sont simulées dans $\mathcal{R}$.

Le fait que MELLP autorise à dupliquer et à effacer des arbres positifs complets ne change absolument rien par rapport au cas d'une boîte puisque les informations multiplicatives et exponentielles sont décorrélées grâce à l'utilisation de piles indépendantes. □

8.38. Exemple

On voit, dans le réseau ci-dessous, la nécessité de pouvoir monter dans une contraction sans information particulière, la nécessité de s'arrêter sur les portes auxiliaires et l'effet de la synchronisation sur le bord d'une boîte.



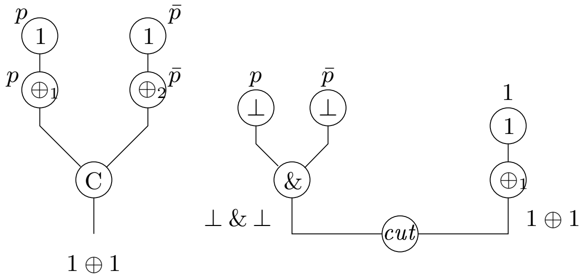
Sans tout cela, il est impossible de distinguer cette preuve de celle qui échange $\mathcal{R}_1$ et $\mathcal{R}_2$ ou bien de celle qui, dans la boîte de droite, échange les nœuds $?w$ et $?d$.

### 8.3.2 Connecteurs additifs

Deux solutions s'offrent alors à nous pour ajouter les connecteurs additifs. On peut utiliser les poids additifs et la méthode de la section 8.1 en l'adaptant aux exponentielles comme dans [Lau01]. La deuxième solution consiste à utiliser *implicitement* la traduction multiplicative de LLP dans MELLP (proposition 4.30), ainsi :

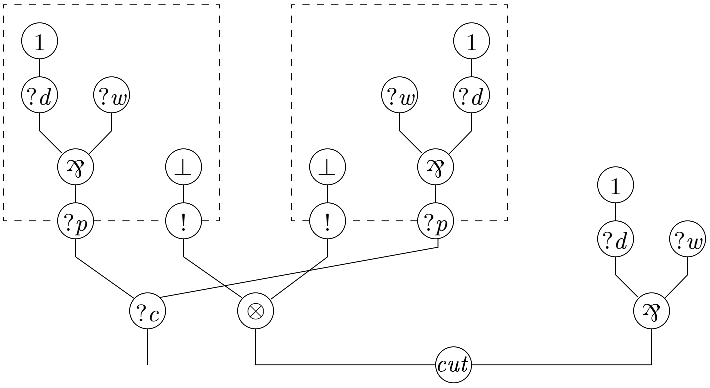
– un nœud & a les mêmes transitions qu'un nœud $\otimes$ situé sous deux nœuds !,
– un nœud $\oplus$ a les mêmes transitions qu'un nœud $\parr$ situé sous un nœud $?d$ et un nœud $?w$,
– un nœud C a les mêmes transitions qu'un nœud $?c$ situé sous deux nœuds $?p$,
– un nœud $\top$ a les mêmes transitions qu'un nœud 1 pour la porte principale et que des nœuds $?w$ pour les portes auxiliaires.

8.39. Exemple
Si on applique cette deuxième procédure au réseau qui correspond à la preuve de calcul des séquents de $\vdash 1 \oplus 1$ donnée au début de la section 8.1.4 pour $i = 1$ :




le calcul de la machine nous donne le jeton $(g, \varepsilon, (g\star).\star)$ dans la conclusion, ce qui correspond bien à une preuve de $1 \oplus 1$ utilisant $\oplus_1$.

Le réseau que l'on a implicitement utilisé pour cela, est :




8.40. Remarque : Il est possible de relier la géométrie de l'interaction polarisée avec la sémantique des jeux polarisés dans l'esprit des travaux de P. Baillot [Bai99]. Ainsi, on pourrait montrer que la machine parallèle que nous avons définie pour la GoI polarisée est très étroitement liée aux jeux AJM polarisés.

Bien qu'issus d'une approche différente basée sur l'analyse de la GoI parallèle pour les connecteurs additifs en partant du travail de J.-Y. Girard [Gir95a], les résultats ci-dessus semblent fortement liés à ceux donnés par S. Abramsky et R. Jagadeesan [AJ94].

# Troisième partie

# Interprétation des systèmes classiques

Par logique classique, on entend généralement calcul des séquents $\mathsf{LK}$ de Gentzen. Cependant du point de vue calculatoire, l'élimination des coupures dans ce système est fortement non déterministe et ce n'est pas le propos de cette thèse.

Nous ne considérerons que les systèmes classiques *constructifs* (ou déterministes) dont l'étude et le nombre ont été très importants ces dix dernières années. L'aspect qui nous intéresse est l'extension de l'expressivité algorithmique des preuves que cela apporte par rapport au cas intuitionniste, sans pour autant quitter le domaine du calcul déterministe.

Nous nous intéresserons aux systèmes classiques déterministes ou à des *fragments* déterministes de systèmes plus vastes comme pour $\mathsf{LK}^{tq}$ sans couleur [DJS97, UB99] ou le $\bar{\lambda}\mu\tilde{\mu}$-calcul [CH00], ...

Un certain nombre de résultats que nous présentons ici s'apparentent à ceux obtenus par Danos-Joinet-Schellinx [DJS97] pour l'analyse des systèmes classiques en logique linéaire. Les nouveautés principales décrites ici sont d'une part l'introduction de traductions utilisant moins de connecteurs exponentiels que dans le cas de $\mathsf{LK}^{tq}$ et d'autre part la clarification des liens avec l'appel par nom et l'appel par valeur. On verra au chapitre 13, que ce raffinement sur les exponentiels est significatif. De plus nous préciserons les rapports entre les différents systèmes classiques grâce à cette analyse dans $\mathsf{LLP}$.

# Chapitre 9

# Traductions

Les systèmes classiques que nous allons interpréter dans $\mathsf{LLP}$ sont de deux catégories : les systèmes polarisés ($\mathsf{LC}$ et $\mathsf{LK}^{\eta}_{\text{pol}}$) qui possèdent explicitement deux classes de formules (positives et négatives) et les systèmes déterministes non polarisés ($\lambda\mu$-calcul, $\lambda c$-calcul, $\lambda\kappa$-calcul, $\mathsf{LKT}$, $\mathsf{LKQ}$, ...). Pour les premiers, les traductions respectent tout simplement les polarités en traduisant les formules classiques positives par des formules linéaires positives et de même pour les négatives, on parlera de *traduction polarisée*. En ce qui concerne les autres systèmes qui ne possèdent qu'une classe de formules, deux traductions sont possibles : traduire toutes les formules par des formules positives (*traduction positive*), ce qui correspond à l'appel par valeur, ou bien toutes par des formules négatives (*traduction négative*), ce qui correspond à l'appel par nom.

## 9.1 Traduction polarisée

Les systèmes classiques que l'on est intéressé à traduire ici utilisent deux conjonctions (multiplicative $\wedge_m$ et additive $\wedge_a$) et deux disjonctions (multiplicative $\vee_m$ et additive $\vee_a$), correspondant à celles de la logique linéaire.

9.1. Définition (Formules positives et négatives)
On appelle *formule positive* toute formule classique dont le connecteur principal est $X$, $V$, $F$, $\wedge_m$, $\vee_a$ ou $\exists$.

On appelle *formule négative* toute formule classique dont le connecteur principal est $\neg X$, $\neg V$, $\neg F$, $\wedge_a$, $\vee_m$ ou $\forall$.

9.2. Définition (Traduction polarisée des formules et séquents)
La *traduction polarisée* $(.)^\bullet$ des formules traduit les formules classiques positives (resp. négatives) en formule polarisée positive (resp. négative). Elle est définie sur les formules atomiques par :

$$\begin{array}{ccc} X & \rightsquigarrow & X \\ \neg X & \rightsquigarrow & X^\perp \\ V & \rightsquigarrow & 1 \\ \neg V & \rightsquigarrow & \perp \\ F & \rightsquigarrow & 0 \\ \neg F & \rightsquigarrow & \top \end{array}$$

et selon la polarité des sous-formules pour les autres connecteurs :

| $A$ | $B$ | $(A \wedge_m B)^\bullet$ | $(A \wedge_a B)^\bullet$ | $(A \vee_m B)^\bullet$ | $(A \vee_a B)^\bullet$ | $(\forall X A)^\bullet$ | $(\exists X A)^\bullet$ |
|---|---|---|---|---|---|---|---|
| $+$ | $+$ | $A^\bullet \otimes B^\bullet$ | $?A^\bullet \,\&\, ?B^\bullet$ | $?A^\bullet \parr ?B^\bullet$ | $A^\bullet \oplus B^\bullet$ | $\forall X ?A^\bullet$ | $\exists X A^\bullet$ |
| $+$ | $-$ | $A^\bullet \otimes !B^\bullet$ | $?A^\bullet \,\&\, B^\bullet$ | $?A^\bullet \parr B^\bullet$ | $A^\bullet \oplus !B^\bullet$ | | |
| $-$ | $+$ | $!A^\bullet \otimes B^\bullet$ | $A^\bullet \,\&\, ?B^\bullet$ | $A^\bullet \parr ?B^\bullet$ | $!A^\bullet \oplus B^\bullet$ | $\forall X A^\bullet$ | $\exists X !A^\bullet$ |
| $-$ | $-$ | $!A^\bullet \otimes !B^\bullet$ | $A^\bullet \,\&\, B^\bullet$ | $A^\bullet \parr B^\bullet$ | $!A^\bullet \oplus !B^\bullet$ | | |

Le séquent classique $\vdash \Gamma, \Delta$, où $\Gamma$ ne contient que des formules positives et $\Delta$ que des formules négatives, est traduit par $\vdash ?\Gamma^{\bullet\perp}, \Delta^\bullet$.

## 9.2 Traduction positive

9.3. Définition (Traduction positive des formules et séquents)
La *traduction positive* $(.)^+$ des formules est définie par :

$$\begin{array}{ccc}
X & \rightsquigarrow & X \\
V & \rightsquigarrow & 1 \\
F & \rightsquigarrow & 0 \\
\neg A & \rightsquigarrow & !(A^+)^\perp \\
A \wedge B & \rightsquigarrow & A^+ \otimes B^+ \\
A \vee B & \rightsquigarrow & A^+ \oplus B^+ \\
\forall X A & \rightsquigarrow & !\forall X ?A^+ \\
\exists X A & \rightsquigarrow & \exists X A^+ \\
A \rightarrow B & \rightsquigarrow & !(A^+ \multimap ?B^+) \\
A - B & \rightsquigarrow & A^+ \otimes !B^{+\perp}
\end{array}$$

Le séquent classique $\Gamma \vdash \Delta$ est traduit par $\vdash \Gamma^{+\perp}, ?\Delta^+$.

## 9.3 Traduction négative

9.4. Définition (Traduction négative des formules et séquents)
La *traduction négative* $(.)^-$ des formules est définie par :

$$\begin{array}{ccc}
X & \rightsquigarrow & X^\perp \\
V & \rightsquigarrow & \top \\
F & \rightsquigarrow & \perp \\
\neg A & \rightsquigarrow & ?(A^-)^\perp \\
A \wedge B & \rightsquigarrow & A^- \,\&\, B^- \\
A \vee B & \rightsquigarrow & A^- \parr B^- \\
\forall X A & \rightsquigarrow & \forall X A^- \\
\exists X A & \rightsquigarrow & ?\exists X !A^- \\
A \rightarrow B & \rightsquigarrow & !A^- \multimap B^- \\
A - B & \rightsquigarrow & ?(!A^- \otimes B^{-\perp})
\end{array}$$

Le séquent classique $\Gamma \vdash \Delta$ est traduit par $\vdash ?(\Gamma^-)^\perp, \Delta^-$.

# Chapitre 10

# Systèmes classiques polarisés

Nous débutons l'étude des traductions de systèmes classiques dans LLP, par les systèmes les plus proches puisqu'ils possèdent une polarisation explicite. En particulier, LC est l'«ancêtre» de LLP.

## 10.1 LC

Le système de logique classique déterministe LC, pour lequel est apparue l'idée de polarisation, a été introduit par J.-Y. Girard [Gir91a] puis étendu au second ordre par M. Quatrini [Qua96]. Il s'agit d'un des premiers systèmes qui donne une version déterministe de LK, donc dont on puisse espérer donner une interprétation calculatoire séquentielle.

### 10.1.1 Formules

Les formules de LC sont construites à partir des atomes $X$ et des constantes $V$ et $F$ (tous les trois positifs) à l'aide des connecteurs $\wedge$, $\vee$, $\neg$, $\exists$ et $\forall$. La *polarité* des formules non atomiques est donnée par :

| $A$ | $B$ | $A \wedge B$ | $A \vee B$ | $\neg A$ | $\exists X A$ | $\forall X A$ |
|---|---|---|---|---|---|---|
| $+$ | $+$ | $+$ | $+$ | $-$ | $+$ | $-$ |
| $+$ | $-$ | $+$ | $-$ | | | |
| $-$ | $+$ | $+$ | $-$ | $+$ | $+$ | $-$ |
| $-$ | $-$ | $-$ | $-$ | | | |

### 10.1.2 Règles

Dans les règles qui suivent, les séquents sont de la forme $\vdash \Gamma; \Pi$ ou $\Gamma$ (de même que $\Delta$ dans les règles) est un multi-ensemble de formules quelconques et $\Pi$ (le *bénitier*) est soit vide soit constitué d'une unique formule positive.

$$\frac{}{\vdash \neg P; P}\,ax \qquad \frac{\vdash \Gamma; P \qquad \vdash \neg P, \Delta; \Pi}{\vdash \Gamma, \Delta; \Pi}\,cut_p \qquad \frac{\vdash \Gamma, N; \qquad \vdash \neg N, \Delta; \Pi}{\vdash \Gamma, \Delta; \Pi}\,cut_n$$

$$\frac{\vdash \Gamma; P \qquad \vdash \Delta; Q}{\vdash \Gamma, \Delta; P \wedge Q}\,\wedge \qquad \frac{\vdash \Gamma; P \qquad \vdash \Delta, N;}{\vdash \Gamma, \Delta; P \wedge N}\,\wedge \qquad \frac{\vdash \Gamma, M; \qquad \vdash \Delta; Q}{\vdash \Gamma, \Delta; M \wedge Q}\,\wedge$$

$$\frac{\vdash \Gamma, M; \Pi \qquad \vdash \Gamma, N; \Pi}{\vdash \Gamma, M \wedge N; \Pi}\,\wedge$$

$$\frac{\vdash \Gamma, A, B; \Pi}{\vdash \Gamma, A \vee B; \Pi} \vee \quad A \vee B \text{ négative} \qquad \frac{\vdash \Gamma; P}{\vdash \Gamma; P \vee Q} \vee \qquad \frac{\vdash \Gamma; Q}{\vdash \Gamma; P \vee Q} \vee$$

$$\frac{}{\vdash; V} V \qquad \frac{}{\vdash \Gamma, \neg F; \Pi} \neg F$$

$$\frac{\vdash \Gamma; P}{\vdash \Gamma, P;} de \qquad \frac{\vdash \Gamma; \Pi}{\vdash \Gamma, A; \Pi} wk \qquad \frac{\vdash \Gamma, A, A; \Pi}{\vdash \Gamma, A; \Pi} co$$

$$\frac{\vdash \Gamma, A; \Pi}{\vdash \Gamma, \forall X A; \Pi} \forall \quad X \notin \Gamma, \Pi \qquad \frac{\vdash \Gamma, N[^P/_X];}{\vdash \Gamma; \exists X N} \exists \qquad \frac{\vdash \Gamma; P[^Q/_X]}{\vdash \Gamma; \exists X P} \exists$$

### 10.1.3 Traduction dans LLP

Grâce aux polarités, le système LC ne nécessite pas de mentionner explicitement si un connecteur est multiplicatif ou additif, on a cependant la correspondance suivante :

| | $A \wedge B$ | $A \vee B$ |
|---|---|---|
| $+$ | $A \wedge_m B$ | $A \vee_a B$ |
| $-$ | $A \wedge_a B$ | $A \vee_m B$ |

On peut alors traduire les formules de LC en formules polarisées en utilisant la traduction $(.)^\bullet$ donnée à la section 9.1. En effet on peut exprimer les règles de LLP avec un bénitier ($\Gamma$ et $\Delta$ ne contenant que des formules négatives) en utilisant (et en intégrant) le lemme 4.9 :

$$\frac{}{\vdash N, N^\perp} ax \qquad \frac{\vdash \Gamma, N, \Pi \qquad \vdash N^\perp, \Delta}{\vdash \Gamma, \Delta, \Pi} cut$$

$$\frac{\vdash \Gamma, N, M, \Pi}{\vdash \Gamma, N \parr M, \Pi} \parr \qquad \frac{\vdash \Gamma, P \qquad \vdash \Delta, Q}{\vdash \Gamma, \Delta, P \otimes Q} \otimes$$

$$\frac{\vdash \Gamma, N, \Pi \qquad \vdash \Gamma, M, \Pi}{\vdash \Gamma, N \,\&\, M, \Pi} \& \qquad \frac{\vdash \Gamma, P}{\vdash \Gamma, P \oplus Q} \oplus_1 \qquad \frac{\vdash \Gamma, Q}{\vdash \Gamma, P \oplus Q} \oplus_2$$

$$\frac{\vdash \Gamma, N}{\vdash \Gamma, !N} ! \qquad \frac{\vdash \Gamma, P}{\vdash \Gamma, ?P} ?d \qquad \frac{\vdash \Gamma, \Pi}{\vdash \Gamma, N, \Pi} ?w \qquad \frac{\vdash \Gamma, N, N, \Pi}{\vdash \Gamma, N, \Pi} ?c$$

$$\frac{}{\vdash \Gamma, \top, \Pi} \top \qquad \frac{\vdash \Gamma, \Pi}{\vdash \Gamma, \bot, \Pi} \bot \qquad \frac{}{\vdash 1} 1$$

$$\frac{\vdash \Gamma, N, \Pi}{\vdash \Gamma, \forall X N, \Pi} \forall \quad X \notin \Gamma, \Pi \qquad \frac{\vdash \Gamma, P[^Q/_X]}{\vdash \Gamma, \exists X P} \exists$$

On étend la traduction $(.)^\bullet$ aux séquents avec bénitier en traduisant le séquent de LC $\vdash \Gamma, \Delta; \Pi$, où $\Gamma$ ne contient que des formules positives et $\Delta$ que des formules négatives, par $\vdash ?\Gamma^{\bullet\perp}, \Delta^\bullet, \Pi^\bullet$ (qui est bien un séquent de LLP puisqu'il contient au plus une formule positive).

10.1. Proposition (Simulation)
*À travers la traduction $(.)^\bullet$, la réduction de* LLP *simule la réduction de* LC.

Démonstration : La procédure d'élimination des coupures de LC donnée dans [Gir91a] n'est pas décrite en détails et il serait fastidieux de le faire ici d'autant plus que d'après le résultat que nous sommes entrain de prouver elle se déduit facilement de celle de LLP. Nous nous contentons donc de donner une preuve au même niveau de précision que la description de l'élimination des coupures de LC dans [Gir91a].

- Si la coupure à éliminer est une $cut_n$, elle est traduite par une coupure exponentielle, la réduction dans LLP commence donc par dupliquer la traduction de la preuve de $\vdash \Gamma, N$; si $\neg N$ a été contractée jusqu'à ce que l'une des occurrences de $\neg N$ soit introduite par une règle de déréliction et la coupure devient une coupure non exponentielle. C'est exactement la définition de l'élimination de cette coupure dans LC.
- Si la coupure à éliminer est une $cut_p$, elle est traduite par une coupure non exponentielle et, dans LLP, la preuve de la formule positive est dupliquée si la formule négative coupée a subi des règles de contraction et ceci jusqu'à ce que l'une des occurrences de cette formule négative soit conclusion d'une règle logique. C'est exactement ce qui se passe dans LC. □

Ce résultat autorise à utiliser les réseaux polarisés comme syntaxe pour LC, ce qui résout un problème ouvert posé par Girard :

> *Trouver une meilleure syntaxe (qui serait à* LC *ce que le $\lambda$-calcul typé est à* LJ*) pour la normalisation* [...] *Des réseaux de preuve pourraient être la solution, l'absence de réseaux pour la logique linéaire complète pouvant être compensée par le fait que seules certaines configurations linéaires sont utilisées.* (Girard [Gir91a])

Pour être précis, le fait que l'on ait dû restreindre LLP à taLLP pour pouvoir éliminer correctement les coupures dans les réseaux nécessite d'ajouter la même contrainte sur LC : le bénitier doit être vide dans les règles $\wedge$ sur des formules négatives et dans les règles $\neg F$. Le sous-système obtenu contenant toujours la traduction de LK dans LC décrite dans [Gir91a] puisqu'il s'agit de traduire le séquent $\vdash \Gamma$ de LK par $\vdash \Gamma;$ dans LC, l'usage des séquents avec bénitier est donc réduit aux étapes intermédiaires et est parfaitement compatible avec cette contrainte.

10.2. Corollaire
*Tout modèle dénotationnel de* LLP *fournit un modèle dénotationnel de* LC.

### 10.1.4 Traduction inverse

Le peu de différence entre LC et LLP nous permet de définir une traduction inverse $(.)^\circ$ de LLP dans LC. Il suffit de préciser comment on traduit les formules qui ne sont pas l'image par $(.)^\bullet$ d'une formule de LC :

$$\begin{array}{ccc} !N \otimes !M & \rightsquigarrow & (N^\circ \wedge V) \wedge (M^\circ \wedge V) \\ ?P \parr ?Q & \rightsquigarrow & (P^\circ \vee \neg V) \vee (Q^\circ \vee \neg V) \\ !N & \rightsquigarrow & N^\circ \wedge V \\ ?P & \rightsquigarrow & P^\circ \vee \neg V \end{array}$$

Le séquent $\vdash \mathcal{N}, P$ de LLP est traduit par $\vdash \mathcal{N}^\circ; P^\circ$ dans LC.

Si $A$ est une formule de LC, on a ainsi $(A^\bullet)^\circ = A$. Réciproquement si $B$ est une formule de LLP, on n'a pas nécessairement d'égalité mais tout de même $(B^\circ)^\bullet \simeq B$.

En appliquant cette traduction aux preuves, ce qui supprime les règles !, on obtient $(\pi^\bullet)^\circ = \pi$. Réciproquement $(\pi^\circ)^\bullet$ et $\pi$ ne diffèrent que par la position de certaines règles ! et des applications de $\otimes 1$ et $\parr \bot$.

10.3. Exemple
Concernant les formules, on a $((!N \otimes !M)^\circ)^\bullet = (!N \otimes 1) \otimes (!M \otimes 1) \simeq !N \otimes !M$.

Et pour les preuves :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash P, P^\perp}\,ax}{\vdash ?P, P^\perp}\,?d}{\vdash ?P, !P^\perp}\,!}{\vdash N, ?P, !P^\perp}\,?w \qquad \dfrac{}{\vdash Q, Q^\perp}\,ax}{\vdash N, ?P, !P^\perp \otimes Q, Q^\perp}\,\otimes \quad \xrightarrow{(.)^\circ} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \neg P; P}\,ax}{\vdash \neg P, P;}\,de}{\vdash \neg P, P, \neg V;}\,wk}{\vdash \neg P, P \vee \neg V;}\,\vee}{\vdash N, \neg P, P \vee \neg V;}\,wk \qquad \dfrac{}{\vdash \neg Q; Q}\,ax}{\vdash N, P \vee \neg V, \neg Q; (\neg P) \wedge Q}\,\wedge$$

$$\text{puis} \quad \xrightarrow{(.)^\bullet} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash P, P^\perp}\,ax}{\vdash ?P, P^\perp}\,?d}{\vdash ?P, \perp, P^\perp}\,\perp}{\vdash ?P \parr \perp, P^\perp}\,\parr}{\vdash N, ?P \parr \perp, P^\perp}\,?w}{\vdash N, ?P \parr \perp, !P^\perp}\,! \qquad \dfrac{}{\vdash Q, Q^\perp}\,ax}{\vdash N, ?P \parr \perp, !P^\perp \otimes Q, Q^\perp}\,\otimes$$

qui induit une commutation de règles et l'introduction d'un $\parr \perp$.

10.4. Corollaire
*Tout modèle dénotationnel de* LC *fournit un modèle dénotationnel de* LLP.

Démonstration : On note $[\![.]\!]_{\mathsf{LC}}$ l'interprétation dans le modèle de LC et on interprète les preuves de LLP par $[\![\pi]\!]_{\mathsf{LLP}} = [\![\pi^\circ]\!]_{\mathsf{LC}}$. Soit $\pi$ une preuve de LLP de forme normale $\pi_0$, si $\pi = \pi'^\bullet$ et si $\pi'_0$ est la forme normale de $\pi'$ dans LC, alors $[\![\pi]\!]_{\mathsf{LLP}} = [\![\pi^\circ]\!]_{\mathsf{LC}} = [\![(\pi'^\bullet)^\circ]\!]_{\mathsf{LC}} = [\![\pi']\!]_{\mathsf{LC}} = [\![\pi'_0]\!]_{\mathsf{LC}} = [\![(\pi_0'^\bullet)^\circ]\!]_{\mathsf{LC}} = [\![\pi_0'^\bullet]\!]_{\mathsf{LLP}}$ or, par simulation de la réduction de LC, $\pi_0'^\bullet$ est une forme normale de $\pi$ donc $\pi_0 = \pi_0'^\bullet$ par confluence. On en conclut que $[\![\pi]\!]_{\mathsf{LLP}} = [\![\pi_0]\!]_{\mathsf{LLP}}$.
Si $\pi$ n'est pas de la forme $\pi'^\bullet$, alors $(\pi^\circ)^\bullet$ et $\pi$ diffèrent par la position de certaines règles ! (qui sont réversibles) et par des isomorphismes liés aux constantes mais ceci est identifié par la sémantique et on est ramené au cas précédent. □

## 10.2 $\mathsf{LK}^\eta_{\mathrm{pol}}$

### 10.2.1 Le système

$\mathsf{LK}^\eta_{\mathrm{pol}}$ est le sous-système polarisé de $\mathsf{LK}^{\mathrm{tq}}$ vérifiant la $\eta$-contrainte : «toute formule positive *active* dans une règle logique est *principale*».

Tout comme le bénitier de LC, la $\eta$-contrainte est une contrainte de *focalisation* [And90], cependant elle est un peu plus forte que celle donnée par le bénitier (exemple 10.5).

10.5. Exemple
On considère la preuve suivante dans LC :

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \neg X; X}\,ax \qquad \dfrac{}{\vdash \neg Y; Y}\,ax}{\vdash \neg X, \neg Y; X \wedge Y}\,\wedge}{\vdash \neg X \vee \neg Y; X \wedge Y}\,\vee}{\vdash \neg X \vee \neg Y; (X \wedge Y) \vee Z}\,\vee$$

Sa traduction naturelle dans $\mathsf{LK}^{\mathrm{tq}}$ est :

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \neg X, X}\,ax \qquad \dfrac{}{\vdash \neg Y, Y}\,ax}{\vdash \neg X, \neg Y, X \wedge_m Y}\,\wedge_m}{\vdash \neg X \vee_m \neg Y, X \wedge_m Y}\,\vee_m}{\vdash \neg X \vee_m \neg Y, (X \wedge_m Y) \vee_a Z}\,{\vee_a}^1$$

mais elle ne vérifie pas la $\eta$-contrainte puisque la formule positive $X \wedge_m Y$ est active dans la règle ${\vee_a}^1$ et présente dans le contexte de la règle $\vee_m$ qui précède. La preuve du séquent $\vdash \neg X \vee_m \neg Y, (X \wedge_m Y) \vee_a Z$ dans $\mathsf{LK}^{\eta}_{\mathrm{pol}}$ est :

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \neg X, X}\,ax \qquad \dfrac{}{\vdash \neg Y, Y}\,ax}{\vdash \neg X, \neg Y, X \wedge_m Y}\,\wedge_m}{\vdash \neg X, \neg Y, (X \wedge_m Y) \vee_a Z}\,{\vee_a}^1}{\vdash \neg X \vee_m \neg Y, (X \wedge_m Y) \vee_a Z}\,\vee_m$$

elle correspond à la preuve de $\mathsf{LC}$ :

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \neg X; X}\,ax \qquad \dfrac{}{\vdash \neg Y; Y}\,ax}{\vdash \neg X, \neg Y; X \wedge Y}\,\wedge}{\vdash \neg X, \neg Y; (X \wedge Y) \vee Z}\,\vee}{\vdash \neg X \vee \neg Y; (X \wedge Y) \vee Z}\,\vee$$

La traduction de $\mathsf{LK}^{\eta}_{\mathrm{pol}}$ dans $\mathsf{LL}$ a été étudiée par Danos-Joinet-Schellinx via le P-plongement. Cependant cette traduction nécessite l'introduction de coupures. Afin de résoudre ce problème, M. Quatrini et L. Tortora de Falco ont introduit la $\rho$-contrainte dans le système $\mathsf{LK}^{\eta,\rho}_{\mathrm{pol}}$ [QTdF96] ce qui, en restreignant le système de départ, permet d'obtenir que le P-plongement n'introduise plus de coupures (les cas délicats sont éliminés). $\mathsf{LLP}$ permet une autre approche, en effet on constate que l'image du P-plongement est contenue dans $\mathsf{LL}_{\mathrm{pol}}$ et plutôt que de contraindre le système de départ, il est également possible d'étendre le système d'arrivée en passant de $\mathsf{LL}_{\mathrm{pol}}$ à $\mathsf{LLP}$. Nous présentons tout d'abord cette traduction de $\mathsf{LK}^{\eta}_{\mathrm{pol}}$ dans $\mathsf{LLP}$, la traduction de $\mathsf{LK}^{\eta,\rho}_{\mathrm{pol}}$ dans $\mathsf{LL}_{\mathrm{pol}}$ apparaissant ensuite comme un cas particulier même si historiquement les résultats ont été donnés dans l'ordre inverse.

Nous donnons ici les principaux résultats de [LQTdF00].

### 10.2.2 Traduction dans LLP

Grâce à la $\eta$-contrainte, la traduction $(.)^\bullet$ (qui correspond au P-plongement [DJS97]) permet de traduire $\mathsf{LK}^{\eta}_{\mathrm{pol}}$ dans $\mathsf{LLP}$.

10.6. Proposition (Simulation)
*À travers la traduction $(.)^\bullet$, la réduction de $\mathsf{LLP}$ simule la réduction de $\mathsf{LK}^{\eta}_{\mathrm{pol}}$.*

Démonstration : Soit $\pi$ une preuve de $\mathsf{LK}^{\eta}_{\mathrm{pol}}$ qui se réduit par une étape d'élimination des coupures en $\pi'$. On considère les différents types de coupures du $\mathtt{tq}$-protocole (voir annexe C.3) :
- Si la coupure est de type $L$, on obtient une coupure logique dans $\pi^\bullet$ qui se réduit de la même manière.
- Si la coupure est de type $S_2$, par la $\eta$-contrainte, la traduction de la sous-preuve contenant la formule de *couleur* $\mathtt{q}$ est un arbre positif. Dans $\mathsf{LK}^{\eta}_{\mathrm{pol}}$, cette sous-preuve est dupliquée et mise aux feuilles de l'arbre structurel de la formule $\mathtt{t}$, c'est exactement ce qu'il se passe dans le réseau $\pi^\bullet$.

– Si la coupure est de type $S_1$, la formule coupée de couleur $\mathtt{q}$ est traduite par une formule $?P$ et la traduction de la sous-preuve contenant la formule de couleur $\mathtt{t}$ se termine par une règle $!$. Dans $\mathsf{LK}^{\eta}_{\mathrm{pol}}$, cette sous-preuve est dupliquée et mise aux feuilles de l'arbre structurel de la formule $\mathtt{q}$, c'est exactement ce qu'il se passe dans le réseau $\pi^{\bullet}$ puisque que cette sous-preuve est traduite par une boîte. □

10.7. COROLLAIRE
*Tout modèle dénotationnel de* $\mathsf{LLP}$ *fournit un modèle dénotationnel de* $\mathsf{LK}^{\eta}_{\mathrm{pol}}$.

### 10.2.3 Renversement

La $\rho$-contrainte a été introduite par M. Quatrini et L. Tortora de Falco dans [QTdF96], puis détaillée dans [LQTdF00].

10.8. DÉFINITION ($\rho$-CONTRAINTE)
Soit $\pi$ une preuve de $\mathsf{LK}^{\eta}_{\mathrm{pol}}$, $\pi$ est *$\rho$-contrainte* si elle ne contient pas :
- de règle structurelle sur une formule négative non atomique,
- de règle irréversible dont une formule négative *active* est dans un contexte contenant une formule négative non atomique,
- de coupure $S_1$ dont la formule positive n'est pas *principale* dans un axiome et dont la formule négative est dans un contexte contenant une formule négative non atomique.

Le sous-système de $\mathsf{LK}^{\eta}_{\mathrm{pol}}$ ne contenant que des preuves $\rho$-contraintes est appelé $\mathsf{LK}^{\eta,\rho}_{\mathrm{pol}}$.

On ne considère ici que des preuves sans quantificateur du second ordre sinon la proposition suivante est violée et la $\rho$-contrainte n'est donc plus appropriée.

10.9. EXEMPLE
La $\rho$-contrainte n'est pas stable par réduction au second ordre :

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\vdash P} \quad \dfrac{\vdots}{\vdash N, \neg X}}{\vdash P \wedge_m N, \neg X}\wedge_m}{\vdash P \wedge_m N, \forall X \neg X}\forall \qquad \dfrac{\dfrac{\vdots}{\vdash Q}}{\vdash \exists X X}\exists}{\vdash P \wedge_m N}\mathit{cut} \qquad \rightarrow \qquad \dfrac{\dfrac{\dfrac{\vdots}{\vdash P} \quad \dfrac{\vdots}{\vdash N, \neg Q}}{\vdash P \wedge_m N, \neg Q}\wedge_m \qquad \dfrac{\vdots}{\vdash Q}}{\vdash P \wedge_m N}\mathit{cut}$$

Ceci provient du traitement particulier des atomes par la $\rho$-contrainte.

10.10. PROPOSITION (STABILITÉ)
*Soit $\pi$ une preuve de $\mathsf{LK}^{\eta}_{\mathrm{pol}}$ $\rho$-contrainte, si $\pi$ se réduit en $\pi'$ alors $\pi'$ est $\rho$-contrainte.*

10.11. DÉFINITION ($\eta\rho$-PREUVE)
Soit $N$ une formule négative de $\mathsf{LK}^{\eta}_{\mathrm{pol}}$, la *$\eta\rho$-preuve* associée à $N$ est la preuve de $\vdash \neg N, N$ obtenue (du bas vers le haut) en appliquant les règles d'introduction des connecteurs réversibles de $N$ puis celles des connecteurs duaux dans $\neg N$ puis en recommençant sur les axiomes obtenus.

10.12. EXEMPLE
La $\eta\rho$-preuve de $\vdash \neg X \wedge_m ((\neg Y \vee_m \neg V) \vee_a Z), X \vee_m ((Y \wedge_m V) \wedge_a \neg Z)$ est :

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash \neg X, X$}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash \neg Y, Y$}
\AxiomC{}
\RightLabel{$V$}
\UnaryInfC{$\vdash V$}
\RightLabel{$\wedge_m$}
\BinaryInfC{$\vdash \neg Y, Y \wedge_m V$}
\RightLabel{$\neg V$}
\UnaryInfC{$\vdash \neg Y, \neg V, Y \wedge_m V$}
\RightLabel{$\vee_m$}
\UnaryInfC{$\vdash \neg Y \vee_m \neg V, Y \wedge_m V$}
\RightLabel{${\vee_a}^1$}
\UnaryInfC{$\vdash (\neg Y \vee_m \neg V) \vee_a Z, Y \wedge_m V$}
\RightLabel{$\wedge_m$}
\BinaryInfC{$\vdash \neg X \wedge_m ((\neg Y \vee_m \neg V) \vee_a Z), X, Y \wedge_m V$}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash \neg X, X$}
\AxiomC{}
\RightLabel{$ax$}
\UnaryInfC{$\vdash Z, \neg Z$}
\RightLabel{${\vee_a}^2$}
\UnaryInfC{$\vdash (\neg Y \vee_m \neg V) \vee_a Z), \neg Z$}
\RightLabel{$\wedge_m$}
\BinaryInfC{$\vdash \neg X \wedge_m ((\neg Y \vee_m \neg V) \vee_a Z), X, \neg Z$}
\RightLabel{$\wedge_a$}
\BinaryInfC{$\vdash \neg X \wedge_m ((\neg Y \vee_m \neg V) \vee_a Z), X, (Y \wedge_m V) \wedge_a \neg Z$}
\RightLabel{$\vee_m$}
\UnaryInfC{$\vdash \neg X \wedge_m ((\neg Y \vee_m \neg V) \vee_a Z), X \vee_m ((Y \wedge_m V) \wedge_a \neg Z)$}
\end{prooftree}
$$

10.13. Proposition (Complétude)
*Soit $\pi$ une preuve de $\vdash \Gamma$ dans $\mathsf{LK}$, il existe une preuve $\pi^\rho$ de $\vdash \Gamma$ dans $\mathsf{LK}^{\eta,\rho}_{\text{pol}}$.*

Démonstration : Nous donnons juste les éléments dont nous aurons besoin pour la proposition 10.15. Soit $R$ une règle de $\pi$ ne vérifiant pas la $\rho$-contrainte et soit $N$ la formule négative non atomique qui en est la cause, on introduit, après $R$, une coupure entre $N$ et la $\eta\rho$-preuve de $\vdash \neg N, N$ et on élimine cette coupure. On vérifie alors que $N$ n'est plus un obstacle à la $\rho$-contrainte. □

10.14. Proposition (Traduction de $\mathsf{LK}^{\eta,\rho}_{\text{pol}}$ dans $\mathsf{LL}$)
*En modifiant la traduction $(.)^\bullet$ sur les atomes par $X \rightsquigarrow\, !X$, on obtient une traduction de $\mathsf{LK}^{\eta,\rho}_{\text{pol}}$ dans $\mathsf{LL}^+_{\text{pol}}$ qui simule la réduction.*

Les preuves détaillées de ces propositions se trouvent dans [LQTdF00].

10.15. Proposition (Simulation du renversement)
*À travers la traduction $(.)^\bullet$, le* renversement *de $\mathsf{LLP}$ (définition 4.16) simule le renversement de $\mathsf{LK}^{\eta}_{\text{pol}}$.*

$$
\begin{array}{ccc}
\mathsf{LK}^{\eta}_{\text{pol}} & \xrightarrow{(.)^\bullet} & \mathsf{LLP} \\
{\scriptstyle\rho}\big\downarrow & & \big\downarrow{\scriptstyle\rho} \\
\mathsf{LK}^{\eta,\rho}_{\text{pol}} & \xrightarrow{(.)^\bullet} & \mathsf{LL}^+_{\text{pol}}
\end{array}
$$

Démonstration : Soit $\pi$ une preuve de $\mathsf{LK}^{\eta}_{\text{pol}}$, on considère une étape de renversement (comme décrit dans la preuve de la proposition 10.13). La formule $N$ que l'on coupe est négative non atomique donc sa traduction par $(.)^\bullet$ est une formule négative ne commençant pas par ?. Le renversement de $\pi$ consiste à couper $N$ avec la $\eta\rho$-preuve de $\vdash \neg N, N$. Le renversement de $\pi^\bullet$ consiste, lui, à couper la traduction de $N$ avec la preuve de $\vdash\, !N^{\bullet\perp}, N^\bullet$. Par simulation de la réduction (proposition 10.6), l'élimination de la coupure introduite dans $\pi$ donne une preuve $\pi'$ dont la traduction $\pi'^\bullet$ est la preuve obtenue en réduisant la coupure introduite dans $\pi^\bullet$. □

Alors que pour le cas propositionnel on a le choix entre la paire $\mathsf{LK}^{\eta,\rho}_{\text{pol}}/\mathsf{LL}_{\text{pol}}$ et la paire $\mathsf{LK}^{\eta}_{\text{pol}}/\mathsf{LLP}$, lors du passage au second ordre seule la deuxième solution persiste puisque la $\rho$-contrainte n'est plus stable par réduction.

En fait, comme on le verra à la section 13.3, passer de $\mathsf{LK}^{\eta}_{\text{pol}}/\mathsf{LLP}$ à $\mathsf{LK}^{\eta,\rho}_{\text{pol}}/\mathsf{LL}_{\text{pol}}$, par renversement, correspond à effectuer une $\neg\neg$-traduction, peut-on alors dire que $\mathsf{LK}^{\eta,\rho}_{\text{pol}}$ est réellement un système de logique classique ?

# Chapitre 11

# $\lambda\mu$-calcul

Le $\lambda\mu$-calcul [Par92] est une extension du $\lambda$-calcul permettant une interprétation calculatoire de la logique classique. Les définitions de base sont rappelées en annexe D.

L'objectif de ce chapitre est d'étudier les traductions du $\lambda\mu$-calcul dans LLP aussi bien pour l'appel par nom que pour l'appel par valeur. Cette étude étend celle effectuée dans LL pour le $\lambda$-calcul [Dan90, Reg92, Reg94]. En particulier, la traduction des $\lambda\mu$-termes en réseaux polarisés nous permet de généraliser la $\sigma$-équivalence du $\lambda$-calcul qui identifie des termes ne différant que par des informations de séquentialité inutiles.

Par le biais de LLP, on précise les liens entre le $\lambda\mu$-calcul et LC restés longtemps obscurs.

Ce chapitre est basé sur [Lau02b] dont il reprend et étend les résultats. Nous ne considérerons que le fragment multiplicatif exponentiel de LLP (MELLP) qui suffit pour l'étude du $\lambda\mu$-calcul tel qu'il a été introduit par M. Parigot puisque l'on n'utilise que le connecteur implication.

## 11.1 Réseaux polarisés purs

Les réseaux purs, qui sont aux réseaux ce que les $\lambda$-termes purs sont aux $\lambda$-termes typés, ont été introduits dans un cadre multiplicatif et exponentiel [Dan90, Reg92]. Les réseaux polarisés permettent d'étendre cette notion afin de correspondre au $\lambda\mu$-calcul pur (il est également possible d'ajouter les connecteurs additifs).

Pour MELLP, comme pour MELL, l'idée est de quotienter les types par l'équation $O \simeq {!O} \multimap O = {?O^\perp} \parr O$ correspondant à l'équation $D \simeq D \to D$ du $\lambda$-calcul pur. En notant $I = O^\perp$, on obtient alors quatre types : $O$, $I$, $!O$ et $?I$, où $O$ et $?I$ sont négatifs et $I$ et $!O$ sont positifs.

Un réseau polarisé peut être vu comme une décoration d'un réseau polarisé pur par des formules de LLP (respectant les polarités et les ? et ! initiaux).

11.1. Proposition (Confluence)
*La réduction des réseaux polarisés purs est* confluente.

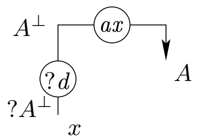
Ce résultat peut se prouver directement ou par traduction dans les réseaux purs [Dan90] de la même manière que pour le corollaire 5.39.

## 11.2 Traduction négative

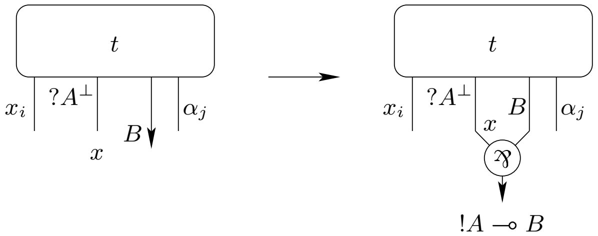
11.2. Définition (Traduction des $\lambda\mu$-termes)
En utilisant la *traduction négative* pour les formules, on traduit un $\lambda\mu$-terme typé (ou son jugement de typage, voir lemme D.3) ayant les variables libres $x_1$, ..., $x_n$ et $\alpha_1$, ..., $\alpha_m$ en un réseau polarisé à

$n+m+1$ conclusions étiquetées par $x_1$, ..., $x_n$, $\alpha_1$, ..., $\alpha_m$ ayant les types correspondants excepté une conclusion qui correspond à la «sortie» du terme et a comme type la traduction de celui du terme. On définit cette traduction par récurrence sur la taille du terme en donnant simultanément la traduction des termes nommés (pour lesquels toutes les conclusions sont étiquetées par une variable) :

$(x)$

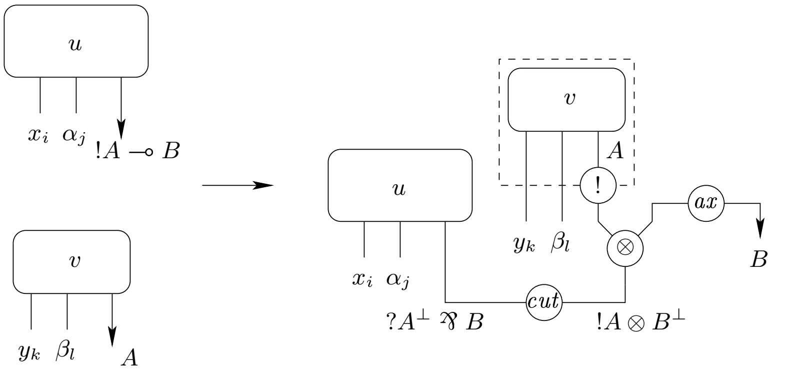



$(\lambda)$



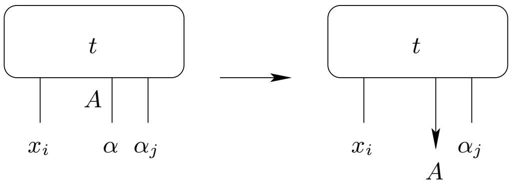

si $x$ n'est pas libre dans $t$, on introduit un nœud $?w$.

$(@)$



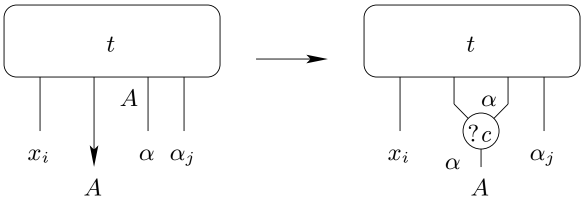

on place des nœuds $?c$ entre les conclusions correspondant à des variables communes à $u$ et $v$.

$(\mu)$




si $\alpha$ n'est pas libre dans $t$, on introduit un nœud $?w$.

$(\alpha)$



si $\alpha$ n'est pas libre dans $t$, il n'y a pas de nœud $?c$ à introduire et on change juste la conclusion distinguée du réseau.

NOTATION : $\rightarrowtail$ représente la conclusion distinguée d'un réseau et $\twoheadrightarrow$ une ancienne conclusion distinguée.

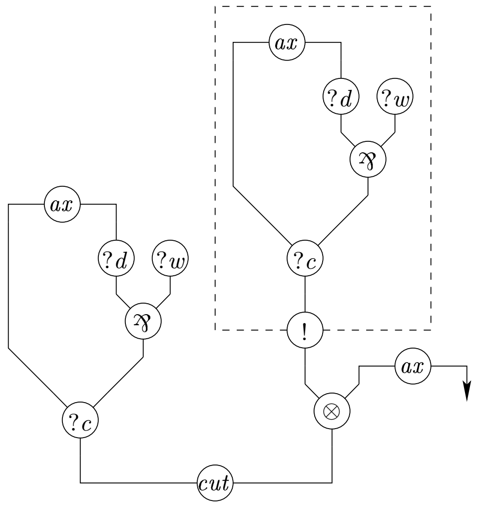
11.3. REMARQUE : Cette traduction peut se décomposer en une traduction du $\lambda\mu$-calcul dans le calcul des séquents LLP suivie de la traduction de ce calcul des séquents en réseaux.

11.4. EXEMPLE
Le réseau de l'exemple 8.38 est la traduction du $\lambda\mu$-terme :

$$(\mu\alpha^{(A\to A\to A)\to A}[\alpha]\lambda f^{A\to A\to A}.\mu\beta^A[\alpha]\lambda d^{A\to A\to A}.((f)\mu\delta^A[\beta]u_1^A)\mu\delta^A[\beta]u_2^A)\lambda x^A.\lambda y^A.x$$

De la même manière, les $\lambda\mu$-termes purs se traduisent en réseaux polarisés purs si on applique l'équation $O \simeq\, !O \multimap O$ aux types négatifs. C'est cette version non typée que l'on considère désormais, le cas typé n'étant alors qu'un cas particulier via les décorations de types appropriées.

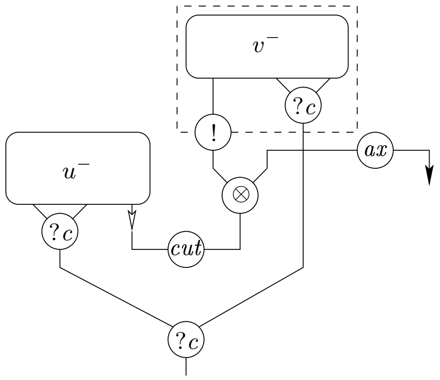
11.5. EXEMPLE
Le $\lambda\mu$-terme $(\mu\alpha[\alpha]\lambda x.\mu\delta[\alpha]x)\mu\alpha[\alpha]\lambda x.\mu\delta[\alpha]x$ est un équivalent pour la $\mu$-réduction du célèbre $(\lambda x.(x)x)\lambda x.(x)x$ et correspond au réseau polarisé pur :



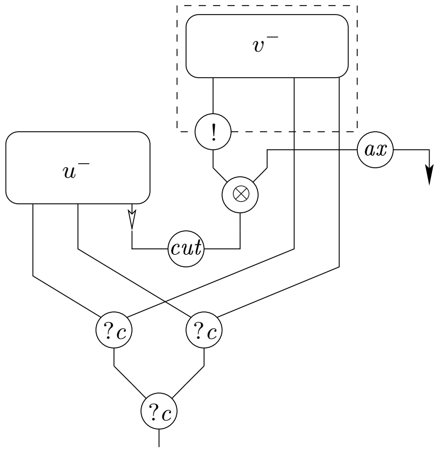

11.6. LEMME (TRADUCTION DE SUBSTITUTION)
- *Soit $t$ un $\lambda\mu$-terme, le réseau $(t[{}^y/_x])^-$ est* structurellement équivalent *au réseau obtenu à partir de $t^-$ en ajoutant un nœud de contraction sous les conclusions correspondant à $x$ et $y$ si $x$ et $y$ sont libres dans $t$ et en n'ajoutant rien sinon.*
- *Soit $t$ un $\lambda\mu$-terme, le réseau $(t[{}^\beta/_\alpha])^-$ est structurellement équivalent au réseau obtenu à partir de $t^-$ en ajoutant un nœud de contraction sous les conclusions correspondant à $\alpha$ et $\beta$ si $\alpha$ et $\beta$ sont libres dans $t$ et en n'ajoutant rien sinon.*

Démonstration : Par récurrence sur la taille de $t$ en considérant chaque construction du $\lambda\mu$-calcul :
- Si $t = x$, $(t[{}^y/_x])^- = y^- = t^-$ et si $t$ est une autre variable $t[{}^y/_x] = t$.
- Si $t = \lambda z.u$, $t[{}^y/_x] = \lambda z.(u[{}^y/_x])$, il suffit d'appliquer l'hypothèse de récurrence à $u$.
- Si $t = (u)v$, $t[{}^y/_x] = (u[{}^y/_x])v[{}^y/_x]$, soit ni $u$ ni $v$ ne contient à la fois $x$ et $y$ libre et on applique simplement l'hypothèse de récurrence soit l'un des deux termes contient les deux variables. On ne considère que le cas où les deux termes contiennent les deux variables, les

autres cas étant des cas particuliers plus simples. Le réseau $((u[{}^{y}/_{x}])v[{}^{y}/_{x}])^{-}$ contient une contraction sur $u^{-}$ et une contraction sur $v^{-}$ à l'intérieur de boîte due à l'application et enfin une contraction entre ces contractions :




ce réseau est structurellement équivalent à :



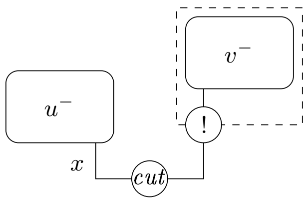

– Si $t = \mu\beta[\alpha]u$, $t[{}^{y}/_{x}] = \mu\beta[\alpha](u[{}^{y}/_{x}])$, il suffit d'appliquer l'hypothèse de récurrence à $u$.

De même pour la $\mu$-substitution avec comme seul cas différent :

– Si $t = \mu\gamma[\gamma']u$, on distingue deux cas selon que $\gamma' = \beta$ ou $\alpha$ ou que $\gamma'$ n'est ni l'une ni l'autre. Dans le premier cas, on a $t[{}^{\beta}/_{\alpha}] = \mu\gamma[\beta]u[{}^{\beta}/_{\alpha}]$ ce qui donne le résultat avec l'équivalence structurelle. Dans le second cas, on a $t[{}^{\beta}/_{\alpha}] = \mu\gamma[\gamma']u[{}^{\beta}/_{\alpha}]$ et on applique l'hypothèse de récurrence. □

11.7. Lemme ($\lambda$-substitution)

*Soient $u$ et $v$ deux $\lambda\mu$-termes n'ayant aucune variable libre en commun. Le réseau :*



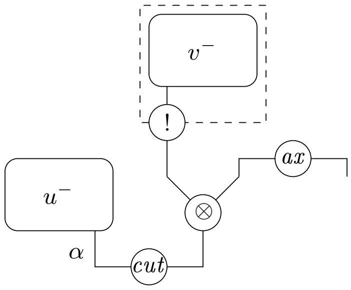

*se réduit en au moins une étape en un réseau structurellement équivalent à $(u[{}^{v}/_{x}])^{-}$.*

Démonstration : Par récurrence sur la taille de $u$ :

– Si $u$ est une variable alors $u = x$ et le résultat est obtenu par une réduction $!/?d$ et une réduction *ax*.

- Si $u = \lambda y.t$, on applique l'hypothèse de récurrence à $t$ ce qui donne le résultat puisque $u[{}^{v}/_{x}] = \lambda y.(t[{}^{v}/_{x}])$.
- Si $u = (t)t'$, on applique l'hypothèse de récurrence à $t$ et à $t'$ s'ils contiennent tous les deux la variable $x$ et sinon uniquement à celui qui la contient. Le réseau se réduit donc en $u[{}^{v}/_{x}] = (t[{}^{v}/_{x}])t'[{}^{v}/_{x}]$ en ajoutant si nécessaire une étape $!/?c$ et une étape $!/!$.
- Si $u = \mu\beta[\alpha]t$, on applique l'hypothèse de récurrence à $t$ ce qui donne le résultat puisque $u[{}^{v}/_{x}] = \mu\beta[\alpha](t[{}^{v}/_{x}])$. □

11.8. Lemme ($\mu$-substitution)
*Soient $u$ et $v$ deux $\lambda\mu$-termes n'ayant aucune variable libre en commun. Le réseau :*




*se réduit (éventuellement en 0 étape) en un réseau structurellement équivalent à $(u[{}^{[\alpha](w)v}/_{[\alpha]w}])^-$.*

Démonstration : Par récurrence sur la taille de $u$ :
- $u$ ne peut pas être une variable.
- Si $u = \lambda x.t$, on applique simplement l'hypothèse de récurrence à $t$.
- Si $u = (t)t'$, on applique l'hypothèse de récurrence à $t$ et à $t'$ s'ils contiennent tous les deux la variable $\alpha$ et sinon uniquement à celui qui la contient. Le réseau se réduit donc en $u[{}^{[\alpha](w)v}/_{[\alpha]w}] = (t[{}^{[\alpha](w)v}/_{[\alpha]w}])t'[{}^{[\alpha](w)v}/_{[\alpha]w}]$ en ajoutant si nécessaire une étape $!/?c$ et une étape $!/!$.
- Si $u = \mu\gamma[\beta]t$, soit $\beta \neq \alpha$ et on applique simplement l'hypothèse de récurrence à $t$, soit $\beta = \alpha$ et il faut en plus appliquer un réduction $!/?c$ si $\alpha \in t$ et enfin si $\alpha \notin t$, il n'y a pas de réduction à faire. □

11.9. Proposition (Simulation)
*Si $u$ se réduit en $v$ (en appel par nom) alors $u^-$ se réduit en $v^-$.*

Démonstration : On considère les deux réductions possibles :
($\beta$) On réduit dans $u$ un rédex $(\lambda x.t)t'$. Pour cela on substitue toutes les variables libres de $t'$ par des variables n'apparaissant pas dans $t$ ce qui donne un terme $t''$, par le lemme 11.7 le sous-réseau de $u^-$ correspondant au rédex se réduit en $t[{}^{t''}/_{x}]$ après une première réduction $\otimes/\parr$. Enfin par le lemme 11.6, on remplace les variables libres de $t''$ pour obtenir $t'$ et le sous-réseau correspondant à $t[{}^{t'}/_{x}]$.
($\mu$) On applique de la même manière les lemmes 11.8 et 11.6 □

## 11.3 Surjectivité

11.10. Définition (Traduction des $\lambda\mu$-termes avec réduction multiplicative)
Soit $u$ un $\lambda\mu$-terme, le réseau $u^\sim$ est obtenu en éliminant dans $u^-$ les coupures multiplicatives.

Cette traduction permet de supprimer de la séquentialité inutile dans les réseaux. Il faut remarquer que puisque l'élimination des coupures multiplicatives se fait en temps linéaire (et en particulier termine toujours), c'est une opération légère et qui ne correspond qu'à une partie très particulière du calcul, la part importante étant réalisée par les autres étapes.

Grâce à cette traduction, on obtient que la traduction d'un terme en forme normale donne un réseau sans coupure car toute coupure du réseau correspond à un rédex du terme. En contrepartie on perd la possibilité d'avoir un résultat de simulation aussi précis que pour la traduction $(.)^{-}$.

11.11. Exemple
La réduction suivante :

$$((\mu\alpha.u)v_1)v_2 \to_\mu (\mu\alpha.u[{}^{[\alpha](w)v_1}/_{[\alpha]w}])v_2 \to_\mu \mu\alpha.u[{}^{[\alpha]((w)v_1)v_2}/_{[\alpha]w}]$$

est réalisée en une seule étape dans le réseau $(((\mu\alpha.u)v_1)v_2)^{\sim}$, la traduction du terme central ne peut donc pas être obtenue par réduction à partir de celle du terme de gauche.

Cependant le résultat de simulation à grands pas reste valable :

11.12. Proposition (Simulation via $(.)^{\sim}$)
*Si $u_0$ est la forme normale de $u$ (en appel par nom) alors $u_0^{\sim}$ est la forme normale de $u^{\sim}$.*

Démonstration : Par la proposition 11.9, $u^{-}$ se réduit en $u_0^{-}$ et par définition de $(.)^{\sim}$, $u_0^{-}$ se réduit en $u_0^{\sim}$, de plus $u_0^{\sim}$ est sans coupure puisque $u_0$ est une forme normale, on en conclut donc que $u_0^{\sim}$ est la forme normale de $u^{-}$ donc celle de $u^{\sim}$ par confluence ($u^{\sim}$ étant un réduit de $u^{-}$). □

11.13. Proposition (Surjectivité)
*Soit $\mathcal{R}$ un réseau polarisé de* MELLP *vérifiant les propriétés suivantes :*

- *exactement une prémisse de chaque nœud $\otimes$ est de type $!N$*
- *exactement une prémisse de chaque nœud $\parr$ est de type $?P$*
- *pas de conclusion positive*
- *pas de nœud $?w$ terminal de type $?P$*
- *uniquement des coupures structurelles*

*il existe un $\lambda\mu$-terme $u$ tel que $u^{\sim} = \mathcal{R}$.*

Exercice : Vous avez bien compris la preuve du théorème 5.19 et la définition 11.2 ? Vous pouvez démontrer la proposition ci-dessus (la solution se trouve dans [Lau02b]).

11.14. Définition (Peignes positifs)
Dans un réseau vérifiant les hypothèses de la proposition précédente, les arbres positifs sont :

- soit des peignes de nœuds $\otimes$ dont toutes les dents sont des boîtes sauf la dernière qui est un nœud $ax$,
- soit une simple boîte et on dit alors que l'arbre positif est *plat*.

11.15. Proposition (Simulation réciproque)
*Si $\mathcal{R}_0$ est la forme normale de $u^{\sim}$ (ou de $u^{-}$) alors la forme normale $u_0$ de $u$ existe et $u_0^{\sim} = \mathcal{R}_0$.*

Démonstration : On considère une réduction finie de $u^{\sim}$ en forme normale qui réduit toutes les coupures multiplicatives après chaque réduction de coupure structurelle, ce qui est toujours possible, si $u^{\sim}$ est normalisable, puisque les réductions multiplicatives font diminuer la taille du réseau.

On raisonne par récurrence sur la longueur de la réduction de $u^{\sim}$. Si $u^{\sim} \to \mathcal{R}_1 \to^* \mathcal{R}_2$ où la première réduction est exponentielle et la seconde est multiplicative, en réduisant dans $u$ le rédex correspondant à la première coupure éliminée, on obtient un terme $u'$ et on a $u'^{\sim} = \mathcal{R}_2$ et par hypothèse de récurrence, $u'$ est normalisable donc $u$ est normalisable.

Soit $u_0$ la forme normale de $u$, par simulation on a $u^{\sim} \to^* u_0^{\sim}$ et donc par confluence des réseaux, $u_0^{\sim} = \mathcal{R}_0$. □

## 11.4 $\sigma$-équivalence

La traduction $(.)^{\sim}$ n'est absolument pas injective et on va désormais chercher à caractériser les termes qui sont traduits par le même réseau en étendant le résultat du $\lambda$-calcul [Reg94].

### 11.4.1 Définition

11.16. DÉFINITION (CONTEXTES ATOMIQUES)
Un *contexte atomique* $C_0$ est un terme obtenu en appliquant l'une des constructions du $\lambda\mu$-calcul à un *trou* au lieu d'un terme :

$$C_0 \quad ::= \quad \lambda y.[\,] \quad | \quad \mu\gamma[\beta][\,] \quad | \quad ([\,])u$$

Un *contexte atomique nommé* $N_0$ se construit de la même manière mais en appliquant les constructions des termes nommés :

$$N_0 \quad ::= \quad [\beta]\lambda y.\mu\gamma.[\,] \quad | \quad [\beta](\mu\gamma.[\,])u$$

La variable $\alpha$ (resp. $x$) est dite *liée* dans $C_0$ si $\alpha \notin C_0[\mu\delta[\alpha]x]$ (resp. $x \notin C_0[x]$). $C_0$ est dit *sans* $\alpha$ (resp. *sans* $x$) si $\alpha$ (resp. $x$) $\notin C_0$ et si $\alpha$ (resp. $x$) n'est pas liée dans $C_0$. On utilise la même terminologie pour un contexte atomique nommé $N_0$.

11.17. DÉFINITION (CONTEXTES)
Un *contexte* $C$ est un terme contenant un *trou* à la place d'un sous-terme, défini de la manière suivante :

$$C \quad ::= \quad [\,] \quad | \quad \lambda y.C \quad | \quad \mu\gamma[\beta]C \quad | \quad (C)u$$

Un *contexte nommé* $N$ est un terme nommé contenant un *trou* à la place d'un sous-terme nommé :

$$N \quad ::= \quad [\,] \quad | \quad [\beta]C[\mu\gamma.[\,]]$$

Les notions de *variable liée dans un contexte*, *contexte sans* $\alpha$ ou *sans* $x$ sont les mêmes que pour les contextes atomiques.

NOTATION : La substitution du trou d'un contexte par un terme est la substitution qui ne considère pas l'$\alpha$-équivalence et autorise la capture de variable. L'égalité de contexte est alors modulo renommage des variable liées telles que le trou n'est pas dans le champ du lieur. Par contre une fois la substitution effectuée, on obtient un $\lambda\mu$-terme et on considère donc l'égalité modulo $\alpha$-équivalence. On a ainsi :

$$\lambda x.[\,] \neq \lambda y.[\,]$$
$$\lambda x.[x] = \lambda x.x \neq \lambda y.[x] = \lambda y.x$$
$$\lambda x.[x] = \lambda x.x = \lambda y.[y] = \lambda y.y$$

$$([\,])\lambda x.(z)x = ([\,])\lambda y.(z)y$$
$$([t])\lambda x.(z)x = (t)\lambda x.(z)x = ([t])\lambda y.(z)y = (t)\lambda y.(z)y$$

Autrement dit $C = C'$ *ssi* pour tout $u$, $C[u] \simeq_\alpha C'[u]$ en tant que termes.

11.18. DÉFINITION ($\sigma$-ÉQUIVALENCE)
La *$\sigma$-équivalence* est la plus petite relation de congruence (*i.e.* préservée par abstraction et application) sur les $\lambda\mu$-termes contenant les équivalences suivantes :

– généralisation de la $\sigma$-équivalence du $\lambda$-calcul aux commutations de $\lambda$-rédex avec toutes les constructions du $\lambda\mu$-calcul,

$$\begin{array}{lclll} ((\lambda x.u)v)w & \simeq_\sigma & (\lambda x.(u)w)v & x \notin w & (\sigma_1) \\ (\lambda x.\lambda y.u)v & \simeq_\sigma & \lambda y(\lambda x.u)v & x \neq y,\ y \notin v & (\sigma_2) \\ (\lambda x\mu\alpha[\beta]u)v & \simeq_\sigma & \mu\alpha[\beta](\lambda x.u)v & \alpha \notin v & (\sigma_3) \end{array}$$

– nouvelles commutations spécifiques au $\lambda\mu$-calcul,

$$\begin{array}{lclll} [\alpha'](\mu\alpha[\beta'](\mu\beta.u)v)w & \simeq_\sigma & [\beta'](\mu\beta[\alpha'](\mu\alpha.u)w)v & \alpha \notin v,\ \beta \notin w & (\mathtt{push/push}) \\ [\alpha']\lambda x.\mu\alpha[\beta']\lambda y.\mu\beta.u & \simeq_\sigma & [\beta']\lambda y.\mu\beta[\alpha']\lambda x.\mu\alpha.u & x \neq y & (\mathtt{pop/pop}) \\ [\alpha'](\mu\alpha[\beta']\lambda x.\mu\beta.u)v & \simeq_\sigma & [\beta']\lambda x.\mu\beta[\alpha'](\mu\alpha.u)v & x \notin v,\ \beta \notin v & (\mathtt{push/pop}) \end{array}$$

avec $\alpha \neq \beta$, $\alpha \neq \beta'$ et $\alpha' \neq \beta$ dans ces trois équations ;

– les $\rho$- et $\theta$-réductions usuelles du $\lambda\mu$-calcul.

$$\begin{array}{lclll} [\beta]\mu\alpha.u & \simeq_\sigma & u[{}^{\beta}/_{\alpha}] & & (\rho) \\ \mu\alpha[\alpha]u & \simeq_\sigma & u & \alpha \notin u & (\theta) \end{array}$$

Ces équations (exceptées $(\rho)$ et $(\theta)$) peuvent également se résumer à l'aide de la notion de contexte atomique :

– commutation d'un $\lambda$-rédex avec un contexte atomique :

$$C_0[(\lambda x.u)v] \quad \simeq_\sigma \quad (\lambda x.C_0[u])v \qquad (\sigma_i)$$

où $C_0$ est sans $x$ et aucune variable libre de $v$ n'est liée dans $C_0$,

– commutation de deux contextes atomiques nommés :

$$N_0[N_0'[u]] \quad \simeq_\sigma \quad N_0'[N_0[u]] \qquad (\mathtt{p/p})$$

où pour tout $\alpha$ (ou $x$) liée dans $N_0$ (resp. $N_0'$), $N_0'$ (resp. $N_0$) est sans $\alpha$ (ou sans $x$).

Ces équations peuvent se justifier opérationnellement en regardant ce à quoi elles correspondent dans la machine de Krivine [Kri92] (voir annexe D.4).

Considérons, par exemple, le cas de l'équation $(\mathtt{push/pop})$ (ce qui devrait éclaircir le choix du nom de cette équation). Supposons que le calcul commence dans l'état $(([\alpha'](\mu\alpha[\beta']\lambda x.\mu\beta.u)v, e), \varepsilon)$ où $e$ contient $(\alpha' = \pi_1)$ et $(\beta' = c :: \pi_2)$ $(\alpha \notin c)$, la machine fait les opérations suivantes :

– restaurer $\pi_1$ ;
– mettre $(v, e)$ sur $\pi_1$ ;
– mémoriser la nouvelle pile $(v, e) :: \pi_1$ avec le nom $\alpha$ dans l'environnement ;
– restaurer la pile $c :: \pi_2$ ;
– dépiler $c$ et la mémoriser avec le nom $x$ dans l'environnement ;
– sauvegarder $\pi_2$ avec le nom $\beta$.

Si on note $e'$ l'état obtenu en ajoutant $(\alpha = (v, e) :: \pi_1)$, $(x = c)$ et $(\beta = \pi_2)$ dans $e$, la machine atteint l'état $((u, e'), \varepsilon)$. On peut vérifier qu'en partant de l'état $(([\beta']\lambda x.\mu\beta[\alpha'](\mu\alpha.u)v, e), \varepsilon)$, la machine passe par les mêmes étapes (juste dans un ordre différent, et avec un environnement légèrement différent associé à $v$ ce qui est sans conséquence puisque $x \notin v$ et $\beta \notin v$).

### 11.4.2 Adéquation

On va montrer que la $\sigma$-équivalence caractérise exactement le quotient sur les $\lambda\mu$-termes réalisé par la traduction $(.)^{\sim}$ dans les réseaux.

Tout d'abord, on généralise la $\sigma$-équivalence à des contextes nommés quelconques :

11.19. Lemme (pop out)
*Si $N$ est sans $x$ et sans $\alpha'$, si $\alpha$ n'est pas liée dans $N$ et si $\alpha' \notin u$, alors :*

$$N[[\alpha]\lambda x.u] \simeq_\sigma [\alpha]\lambda x.\mu\alpha'.N[[\alpha']u]$$

Démonstration : Par récurrence sur $N$ :

- Si $N = [\,]$ :
$$[\alpha]\lambda x.u \simeq_\sigma [\alpha]\lambda x.\mu\alpha'[\alpha']u \qquad \text{par } (\theta)$$
- Si $N = [\beta]\mu\gamma.[\,]$ :
$$\begin{aligned}[\beta]\mu\gamma[\alpha]\lambda x.u &\simeq_\sigma [\beta]\mu\gamma[\alpha]\lambda x.\mu\alpha'[\alpha']u && \text{par } (\theta)\\ &\simeq_\sigma [\alpha]\lambda x.\mu\alpha'[\alpha']u[^\beta/_\gamma] && \text{par } (\rho)\\ &\simeq_\sigma [\alpha]\lambda x.\mu\alpha'[\beta]\mu\gamma[\alpha']u && \text{par } (\rho)\end{aligned}$$
- Si $N = [\beta]\lambda y.C[\mu\gamma.[\,]]$ :
$$\begin{aligned}[\beta]\lambda y.C[\mu\gamma.[\alpha]\lambda x.u] &\simeq_\sigma [\beta]\lambda y.\mu\beta'[\beta']C[\mu\gamma.[\alpha]\lambda x.u] && \text{par } (\theta)\\ &\simeq_\sigma [\beta]\lambda y.\mu\beta'[\alpha]\lambda x.\mu\alpha'[\beta']C[\mu\gamma[\alpha']u] && \text{par récurrence}\\ &\simeq_\sigma [\alpha]\lambda x.\mu\alpha'[\beta]\lambda y.\mu\beta'[\beta']C[\mu\gamma[\alpha']u] && \text{par } (\texttt{pop}/\texttt{pop})\\ &\simeq_\sigma [\alpha]\lambda x.\mu\alpha'[\beta]\lambda y.C[\mu\gamma[\alpha']u] && \text{par } (\theta)\end{aligned}$$
- Si $N = [\beta]\mu\beta'[\gamma']C[\mu\gamma.[\,]]$ :
$$\begin{aligned}[\beta]\mu\beta'[\gamma']C[\mu\gamma[\alpha]\lambda x.u] &\simeq_\sigma [\beta]\mu\beta'[\alpha]\lambda x.\mu\alpha'[\gamma']C[\mu\gamma[\alpha']u] && \text{par récurrence}\\ &\simeq_\sigma [\alpha]\lambda x.\mu\alpha'[\gamma']C[\mu\gamma[\alpha']u][^\beta/_{\beta'}] && \text{par } (\rho)\\ &\simeq_\sigma [\alpha]\lambda x.\mu\alpha'[\beta]\mu\beta'[\gamma']C[\mu\gamma[\alpha']u] && \text{par } (\rho)\end{aligned}$$
- Si $N = [\beta](C[\mu\gamma.[\,]])v$ :
$$\begin{aligned}[\beta](C[\mu\gamma[\alpha]\lambda x.u])v &\simeq_\sigma [\beta](\mu\beta'[\beta']C[\mu\gamma[\alpha]\lambda x.u])v && \text{par } (\theta)\\ &\simeq_\sigma [\beta](\mu\beta'[\alpha]\lambda x.\mu\alpha'[\beta']C[\mu\gamma[\alpha']u])v && \text{par récurrence}\\ &\simeq_\sigma [\alpha]\lambda x.\mu\alpha'[\beta](\mu\beta'[\beta']C[\mu\gamma[\alpha']u])v && \text{par } (\texttt{push}/\texttt{pop})\\ &\simeq_\sigma [\alpha]\lambda x.\mu\alpha'[\beta](C[\mu\gamma[\alpha']u])v && \text{par } (\theta)\end{aligned}$$

□

11.20. Lemme (push out)
*Si $N$ est sans $\alpha'$, si $\alpha$ n'est pas liée dans $N$, si $\alpha' \notin u$ et si aucune des variables libres de $v$ n'est liée dans $N$, alors :*

$$N[[\alpha](u)v] \simeq_\sigma [\alpha](\mu\alpha'.N[[\alpha']u])v$$

Démonstration : Par récurrence sur $N$ :

- Si $N = [\,]$ :
$$[\alpha](u)v \simeq_\sigma [\alpha](\mu\alpha'[\alpha']u)v \qquad \text{par } (\theta)$$
- Si $N = [\beta]\mu\gamma.[\,]$ :
$$\begin{aligned}[\beta]\mu\gamma[\alpha](u)v &\simeq_\sigma [\beta]\mu\gamma[\alpha](\mu\alpha'[\alpha']u)v && \text{par } (\theta)\\ &\simeq_\sigma [\alpha](\mu\alpha'[\alpha']u[^\beta/_\gamma])v && \text{par } (\rho)\\ &\simeq_\sigma [\alpha](\mu\alpha'[\beta]\mu\gamma[\alpha']u)v && \text{par } (\rho)\end{aligned}$$
- Si $N = [\beta]\lambda x.C[\mu\gamma.[\,]]$ :
$$\begin{aligned}[\beta]\lambda x.C[\mu\gamma.[\alpha](u)v] &\simeq_\sigma [\beta]\lambda x.\mu\beta'[\beta']C[\mu\gamma.[\alpha](u)v] && \text{par } (\theta)\\ &\simeq_\sigma [\beta]\lambda x.\mu\beta'[\alpha](\mu\alpha'[\beta']C[\mu\gamma[\alpha']u])v && \text{par récurrence}\\ &\simeq_\sigma [\alpha](\mu\alpha'[\beta]\lambda x.\mu\beta'[\beta']C[\mu\gamma[\alpha']u])v && \text{par } (\texttt{push}/\texttt{pop})\\ &\simeq_\sigma [\alpha](\mu\alpha'[\beta]\lambda x.C[\mu\gamma[\alpha']u])v && \text{par } (\theta)\end{aligned}$$

– Si $N = [\beta]\mu\beta'[\gamma']C[\mu\gamma.[\,]]$ :

$$\begin{aligned}
[\beta]\mu\beta'[\gamma']C[\mu\gamma[\alpha](u)v] &\simeq_\sigma [\beta]\mu\beta'[\alpha](\mu\alpha'[\gamma']C[\mu\gamma[\alpha']u])v && \text{par récurrence}\\
&\simeq_\sigma [\alpha](\mu\alpha'[\gamma']C[\mu\gamma[\alpha']u])v[^{\beta}/_{\beta'}] && \text{par } (\rho)\\
&\simeq_\sigma [\alpha](\mu\alpha'[\beta]\mu\beta'[\gamma']C[\mu\gamma[\alpha']u])v && \text{par } (\rho)
\end{aligned}$$

– Si $N = [\beta](C[\mu\gamma.[\,]])w$ :

$$\begin{aligned}
[\beta](C[\mu\gamma[\alpha](u)v])w &\simeq_\sigma [\beta](\mu\beta'[\beta']C[\mu\gamma[\alpha](u)v])w && \text{par } (\theta)\\
&\simeq_\sigma [\beta](\mu\beta'[\alpha](\mu\alpha'[\beta']C[\mu\gamma[\alpha']u])v)w && \text{par récurrence}\\
&\simeq_\sigma [\alpha](\mu\alpha'[\beta](\mu\beta'[\beta']C[\mu\gamma[\alpha']u])w)v && \text{par } (\texttt{push}/\texttt{push})\\
&\simeq_\sigma [\alpha](\mu\alpha'[\beta](C[\mu\gamma[\alpha']u])w)v && \text{par } (\theta)
\end{aligned}$$

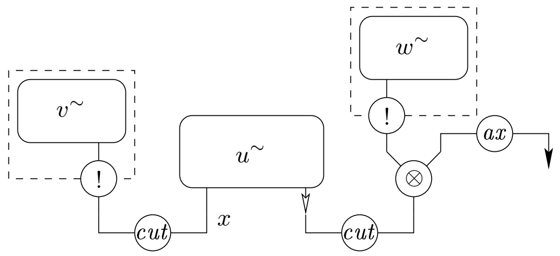
□

11.21. REMARQUE : Dans certains cas particuliers, la $\rho$-réduction peut modifier la traduction d'un terme en ce qui concerne ses liens $?w$ finaux. Par exemple $[\alpha]\mu\delta.u \rightarrow_\rho u$ si $\delta \notin u$ et $([\alpha]\mu\delta.u)^\sim$ est obtenu en ajoutant un nœud $?w$ correspond à $\alpha$ au réseau $u^\sim$. C'est pourquoi on étendra désormais l'*équivalence structurelle* $\simeq_s$ des réseaux en considérant équivalents des réseaux qui diffèrent uniquement par des affaiblissements finaux.

11.22. THÉORÈME (CORRECTION)
*Soient $t$ et $t'$ deux $\lambda\mu$-termes, $t \simeq_\sigma t' \Rightarrow t^\sim \simeq_s t'^\sim$.*

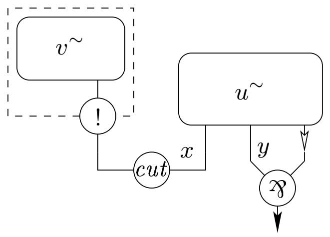
Démonstration : Nous démontrons que les équations de la $\sigma$-équivalence sont bien réalisées par la traduction dans les réseaux.

($\sigma_1$) Les deux réseaux $(((\lambda x.u)v)w)^-$ et $((\lambda x.(u)w)v)^-$ se réduisent tous les deux multiplicativement à :

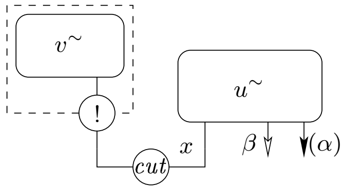



Puisque $(.)^\sim$ s'obtient à partir de $(.)^-$ en éliminant toutes les coupures multiplicatives, en réduisant ce réseau on obtient $(((\lambda x.u)v)w)^\sim$ et $((\lambda x.(u)w)v)^\sim$ qui sont donc égaux.

($\sigma_2$) Les deux réseaux $(((\lambda x.\lambda y.u)v)^\sim$ et $(\lambda y.(\lambda x.u)v)^\sim$ sont égaux à :

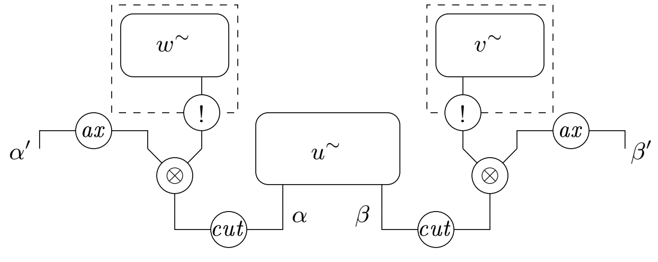



($\sigma_3$) Les deux réseaux $(((\lambda x.\mu\alpha[\beta]u)v)^\sim$ et $(\mu\alpha[\beta](\lambda x.u)v)^\sim$ sont égaux à :

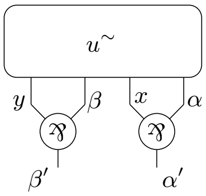


modulo $\simeq_s$ à cause de l'ordre des contractions au-dessus de $\beta$.

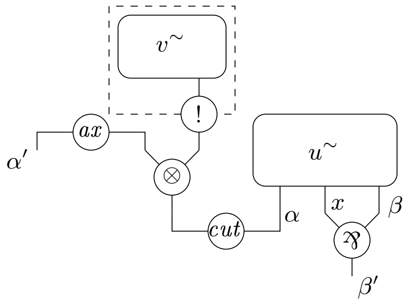
($\texttt{push}/\texttt{push}$) Les deux réseaux $([\alpha'](\mu\alpha[\beta'](\mu\beta.u)v)w)^\sim$ et $([\beta'](\mu\beta[\alpha'](\mu\alpha.u)w)v)^\sim$ sont égaux à :



($\texttt{pop}/\texttt{pop}$) Les deux réseaux $([\alpha']\lambda x.\mu\alpha[\beta']\lambda y.\mu\beta.u)^\sim$ et $([\beta']\lambda y.\mu\beta[\alpha']\lambda x.\mu\alpha.u)^\sim$ sont égaux à :



($\texttt{push}/\texttt{pop}$) Les deux réseaux $([\alpha'](\mu\alpha[\beta']\lambda x.\mu\beta.u)v)^\sim$ et $([\beta']\lambda x.\mu\beta[\alpha'](\mu\alpha.u)v)^\sim$ sont égaux à :



($\rho$) $([\beta]\mu\alpha.u)^\sim$ et $(u[^\beta/_\alpha])^\sim$ sont égaux modulo $\simeq_s$ par le lemme 11.6 et la remarque 11.21.

($\theta$) Le réseau $(\mu\alpha[\alpha]u)^\sim$ est exactement le réseau $u^\sim$ si $\alpha \notin u$. $\square$

La réciproque nécessite quelques lemmes intermédiaires.

11.23. Lemme

*Soit $[\beta]u$ un terme nommé contenant une variable libre $\alpha$ telle que la conclusion de $([\beta]u)^\sim$ correspondant à $\alpha$ ne soit ni porte auxiliaire d'une boîte ni conclusion d'un nœud $?c$. Il existe un contexte $N$ sans $\alpha$ et un $\lambda\mu$-terme $u'$ tels que $[\beta]u = N[[\alpha]u']$ avec $\alpha \notin u'$.*

Démonstration : Si $\beta = \alpha$ alors nécessairement $\alpha \notin u$ sinon la conclusion correspondant à $\alpha$ dans $([\alpha]u)^\sim$ serait conclusion d'un nœud $?c$. Dans ce cas le résultat est prouvé avec $N = [\,]$.

Si $\beta \neq \alpha$, on raisonne par récurrence sur la taille de $u$ :

- Si $u = x$, on a nécessairement $\beta = \alpha$ puisque $\alpha$ est libre dans $[\beta]u$.
- Si $u = \lambda x.u_0$, on peut appliquer l'hypothèse de récurrence à $[\beta]u_0$, donc $[\beta]u_0 = N[[\alpha]u']$ or $N \neq [\,]$ puisque $\beta \neq \alpha$ d'où $N = [\beta]C[\mu\gamma.[\,]]$ et enfin $[\beta]u = [\beta]\lambda x.C[\mu\gamma[\alpha]u']$.
- Si $u = (u_0)v$, $\alpha$ ne peut pas être à la fois libre dans $u_0$ et $v$ puisque la conclusion associée dans $([\alpha]u)^\sim$ n'est pas conclusion d'un nœud $?c$ et $\alpha$ ne peut pas être libre dans $v$ puisque la conclusion associée n'est pas porte auxiliaire d'une boîte. On en déduit que $\alpha$ est libre

dans $u_0$ et pas dans $v$ et que l'on peut appliquer l'hypothèse de récurrence à $[\beta]u_0$, donc $[\beta]u_0 = N[[\alpha]u']$ avec $N = [\beta]C[\mu\gamma.[]]$ (car on a à nouveau $N \neq []$) d'où, enfin $[\beta]u = [\beta](C[\mu\gamma[\alpha]u'])v$.

- Si $u = \mu\gamma[\gamma']u_0$, alors $\gamma \neq \alpha$ puisque $\alpha$ est libre dans $u$. On peut appliquer l'hypothèse de récurrence à $[\gamma']u_0$, d'où $[\gamma']u_0 = N[[\alpha]u']$ et donc $[\beta]u = [\beta]\mu\gamma.N[[\alpha]u']$. □

11.24. LEMME

*Si $t$ est un terme nommé et si $t^\sim$ a un nœud ⅋ terminal (dont la conclusion correspond à la variable libre $\alpha$), alors il existe un terme $t_0$ tel que $t \simeq_\sigma [\alpha]\lambda x.t_0$ et $t_0^\sim$ est le réseau obtenu en supprimant ce nœud ⅋ dans $t^\sim$.*

Démonstration : On raisonne par récurrence sur la taille de $t$ :

- Si $t = [\beta]x$, $t^\sim$ ne contient pas de nœud ⅋.
- Si $t = [\beta]\lambda y.u$, si $\beta = \alpha$ le résultat est prouvé, sinon par hypothèse de récurrence $[\gamma]u \simeq_\sigma [\alpha]\lambda x.t_0$ où $\gamma \notin u$, donc :

$$\begin{aligned} t &= [\beta]\lambda y.u \\ &\simeq_\sigma [\beta]\lambda y.\mu\gamma[\gamma]u && (\theta) \\ &\simeq_\sigma [\beta]\lambda y.\mu\gamma[\alpha]\lambda x.t_0 \\ &\simeq_\sigma [\beta]\lambda y.\mu\gamma[\alpha]\lambda x.\mu\gamma'[\gamma']t_0 && (\theta) \\ &\simeq_\sigma [\alpha]\lambda x.\mu\gamma'[\beta]\lambda y.\mu\gamma[\gamma']t_0 && (\mathsf{pop}/\mathsf{pop}) \end{aligned}$$

- Si $t = [\beta]\mu\gamma.t'$, on applique l'hypothèse de récurrence à $t'$ ce qui donne un terme $t_0$ tel que $t' \simeq_\sigma [\alpha]\lambda x.t_0$ donc :

$$\begin{aligned} t &\simeq_\sigma [\beta]\mu\gamma[\alpha]\lambda x.t_0 \\ &\simeq_\sigma [\alpha]\lambda x.t_0[{}^\beta/_\gamma] && (\rho) \end{aligned}$$

- Si $t = [\beta](u)v_1 \dots v_n$ où $n > 0$ et $u$ n'est pas une application, on considère chaque cas possible pour $u$ :
  - Si $u = x$ alors $t^\sim$ ne contient pas de nœud ⅋ à profondeur 0.
  - Si $u = \lambda y.u'$, par hypothèse de récurrence $[\beta](u')v_2 \dots v_n \simeq_\sigma [\alpha]\lambda x.t_0$, si on choisit $x$ non libre dans $v_1$, on a alors :

$$\begin{aligned} t &= [\beta](\lambda y.u')v_1v_2 \dots v_n \\ &\simeq_\sigma [\beta](\lambda y.(u')v_2 \dots v_n)v_1 && (\sigma_1)^{n-1} \\ &\simeq_\sigma [\beta]\mu\beta[\beta](\lambda y.(u')v_2 \dots v_n)v_1 && (\rho) \\ &\simeq_\sigma [\beta](\lambda y.\mu\beta[\beta](u')v_2 \dots v_n)v_1 && (\sigma_3) \\ &\simeq_\sigma [\beta](\lambda y.\mu\beta[\alpha]\lambda x.t_0)v_1 \\ &\simeq_\sigma [\beta]\mu\beta[\alpha](\lambda y.\lambda x.t_0)v_1 && (\sigma_3) \\ &\simeq_\sigma [\alpha](\lambda y.\lambda x.t_0)v_1 && (\rho) \\ &\simeq_\sigma [\alpha]\lambda x.(\lambda y.t_0)v_1 && (\sigma_2) \end{aligned}$$

  - Si $u = \mu\gamma[\gamma']u'$, si $\beta \neq \alpha$ alors $\gamma \neq \alpha$ et on peut appliquer l'hypothèse de récurrence à $[\gamma']u'$ d'où $[\gamma']u' \simeq_\sigma [\alpha]\lambda x.t_0$ et :

$$\begin{aligned} t &= [\beta](\mu\gamma[\gamma']u')v_1 \dots v_n \\ &\simeq_\sigma [\beta](\mu\gamma[\alpha]\lambda x.t_0)v_1 \dots v_n \\ &\simeq_\sigma [\alpha]\lambda x.\mu\alpha'[\beta](\mu\gamma[\alpha']t_0)v_1 \dots v_n && \text{lemme 11.19} \end{aligned}$$

  et si $\beta = \alpha$, la conclusion distinguée de $((\mu\gamma[\gamma']u')v_1 \dots v_n)^\sim$ est un nœud ⅋, on en déduit que la conclusion distinguée de $(\mu\gamma[\gamma']u')^\sim$ est un nœud ⅋ ou $ax$ pour qu'une réduction multiplicative puisse avoir lieu lors de la traduction de l'application aux $v_i$ sinon on obtiendrait le nœud $ax$ de l'arbre positif formé des $v_i$ comme conclusion distinguée de $((\mu\gamma[\gamma']u')v_1 \dots v_n)^\sim$. Ceci nous permet d'appliquer le lemme 11.23 à $[\gamma']u'$, on obtient $[\gamma']u' = N[[\gamma]u'']$ et :

$$
\begin{aligned}
t &= [\alpha](\mu\gamma[\gamma']u')v_1\dots v_n \\
&\simeq_\sigma [\alpha](\mu\gamma.N[[\gamma]u''])v_1\dots v_n \\
&\simeq_\sigma N[[\alpha](u'')v_1\dots v_n] && \text{lemme 11.20} \\
&\simeq_\sigma N[[\alpha]\lambda x.t_0] && \text{récurrence} \\
&\simeq_\sigma [\alpha]\lambda x.\mu\alpha'.N[[\alpha']t_0] && \text{lemme 11.19}
\end{aligned}
$$

□

11.25. Lemme

*Si $t$ est un terme nommé et si $t^\sim$ n'a pas de nœud négatif terminal mais contient un nœud cut maximal dont l'arbre positif n'est pas plat et à $\alpha$ comme variable associée à la conclusion de son lien ax, il existe $t_0$ et $v$ tels que $t \simeq_\sigma [\alpha](t_0)v$ et $t_0^\sim$ est le réseau obtenu en supprimant la boîte contenant $v^\sim$ dans $t^\sim$ (modulo réduction multiplicative).*

Démonstration : On raisonne par récurrence sur la taille de $t$. La conclusion correspondant à $\alpha$ dans $t^\sim$ étant celle d'un nœud *ax*, on peut appliquer le lemme 11.23, d'où $t = N[[\alpha]t']$ avec soit $N \neq [\,]$ et on peut appliquer l'hypothèse de récurrence à $[\alpha]t'$ ce qui donne $[\alpha]t' \simeq_\sigma [\alpha](t_0)v$ et, par le lemme 11.20, $t \simeq_\sigma [\alpha](\mu\alpha'.N[[\alpha']t_0])v$ ; soit $N = [\,]$ et on est dans l'un des cas suivants :

- Si $t = [\alpha]x$, $t^\sim$ ne contient pas de coupure.
- Si $t = [\alpha]\lambda x.u$, $t^\sim$ a un nœud négatif ⅋ ou $?c$ terminal.
- Si $t = [\alpha]\mu\beta.t'$, il suffit d'appliquer l'hypothèse de récurrence à $t'[^\alpha/_\beta]$.
- Si $t = [\alpha](u)v_1\dots v_n$ où $n > 0$ et $u$ n'est pas une application, on considère chaque cas possible pour $u$ :
  - Si $u = x$, le résultat est prouvé avec $v = v_n$.
  - Si $u = \lambda x.u'$, on a :

$$
\begin{aligned}
t &= [\alpha](\lambda x.u')v_1v_2\dots v_n \\
&\simeq_\sigma [\alpha](\lambda x.(u')v_2\dots v_n)v_1 && (\sigma_1)^{n-1} \\
&\simeq_\sigma [\alpha](\lambda x.\mu\alpha[\alpha](u')v_2\dots v_n)v_1 && (\theta) \\
&\simeq_\sigma [\alpha](\lambda x.\mu\alpha[\alpha](t_0)v)v_1 && \text{récurrence} \\
&\simeq_\sigma [\alpha](\lambda x.(t_0)v)v_1 && (\theta) \\
&\simeq_\sigma [\alpha](\lambda x.t_0)v_1v && (\sigma_1)
\end{aligned}
$$

  - Si $u = \mu\beta[\gamma]u'$, soit le lemme est vérifié pour $v = v_n$, soit comme dans le lemme 11.24, la conclusion de $([\gamma]u')^\sim$ correspondant à $\beta$ est conclusion d'un nœud *ax* ou ⅋ pour qu'il y ait pu avoir réduction multiplicative lors de la traduction et dans ce cas on peut appliquer le lemme 11.23 à $[\gamma]u'$ d'où $[\gamma]u' = N[[\beta]u'']$ et :

$$
\begin{aligned}
t &= [\alpha](\mu\beta[\gamma]u')v_1\dots v_n \\
&\simeq_\sigma [\alpha](\mu\beta.N[[\beta]u''])v_1\dots v_n \\
&\simeq_\sigma N[[\alpha](u'')v_1\dots v_n] && \text{lemme 11.20} \\
&\simeq_\sigma N[[\alpha](t_0)v] && \text{récurrence} \\
&\simeq_\sigma [\alpha](\mu\alpha'.N[[\alpha']t_0])v && \text{lemme 11.20}
\end{aligned}
$$

□

11.26. Lemme

*Si $t$ est un terme nommé et si $t^\sim$ n'a pas de nœud négatif terminal mais contient un nœud cut maximal dont l'arbre positif est plat, il existe $v$ tel que, pour tout $\alpha \in t$ ($\alpha \notin v$), il existe un terme $t_0$ tel que $t \simeq_\sigma [\alpha](\lambda x.t_0)v$ et $[\alpha]t_0^\sim$ est le réseau obtenu en supprimant la coupure et la boîte contenant $v^\sim$ dans $t^\sim$.*

Démonstration : Il convient tout d'abord de constater que la validité de $v$ «pour tout $\alpha \in t$ ($\alpha \notin v$)» n'est absolument pas une contrainte puisque si on obtient le résultat pour un $\alpha$ particulier, on l'obtient automatiquement pour tous les autres :

$$\begin{aligned}
t &\simeq_\sigma [\alpha](\lambda x.t_0)v \\
&\simeq_\sigma [\alpha']\mu\alpha'[\alpha](\lambda x.t_0)v && (\rho) \\
&\simeq_\sigma [\alpha'](\lambda x.\mu\alpha'[\alpha]t_0)v && (\sigma_3)
\end{aligned}$$

On raisonne par récurrence sur la taille de $t$ :

- Si $t = [\beta]x$, $t^\sim$ ne contient pas de coupure.
- Si $t = [\beta]\lambda x.u$, $t^\sim$ a un nœud négatif ⅋ ou $?c$ terminal.
- Si $t = [\beta]\mu\gamma.t'$, il suffit d'appliquer l'hypothèse de récurrence à $t'[^\beta/_\gamma]$.
- Si $t = [\beta](u)v_1 \dots v_n$ où $n > 0$ et $u$ n'est pas une application, on considère chaque cas possible pour $u$ :
  - Si $u = x$, $t^\sim$ ne contient pas de coupure dont l'arbre positif soit plat à profondeur 0.
  - Si $u = \lambda y.u'$, en appliquant $n-1$ fois $(\sigma_1)$ on a $t \simeq_\sigma [\beta](\lambda y.(u')v_2 \dots v_n)v_1$. Soit le lemme est vérifié avec $v = v_1$, soit on applique l'hypothèse de récurrence à $[\gamma](u')v_2 \dots v_n$ avec $\gamma \notin (u')v_2 \dots v_n$ :
  $$\begin{aligned}
  t &\simeq_\sigma [\beta](\lambda y.(u')v_2 \dots v_n)v_1 \\
  &\simeq_\sigma [\beta](\lambda y.\mu\gamma[\gamma](u')v_2 \dots v_n)v_1 && (\theta) \\
  &\simeq_\sigma [\beta](\lambda y.\mu\gamma[\gamma](\lambda x.t_0)v)v_1 && \text{récurrence} \\
  &\simeq_\sigma [\beta](\lambda y.(\lambda x.t_0)v)v_1 && (\theta) \\
  &\simeq_\sigma [\beta](\lambda y.\lambda x.t_0)v_1 v && (\sigma_1) \\
  &\simeq_\sigma [\beta](\lambda x.(\lambda y.t_0)v_1)v && (\sigma_2)
  \end{aligned}$$
  - Si $u = \mu\gamma[\gamma']u'$, soit l'arbre positif plat fait partie du sous-réseau de $t^\sim$ correspondant à $[\gamma']u'^\sim$ et en y appliquant l'hypothèse de récurrence, on obtient :
  $$\begin{aligned}
  t &\simeq_\sigma [\beta](\mu\gamma[\gamma']u')v_1 \dots v_n \\
  &\simeq_\sigma [\beta](\mu\gamma[\gamma'](\lambda x.t_0)v)v_1 \dots v_n && \text{récurrence} \\
  &\simeq_\sigma [\beta](\lambda x.\mu\gamma[\gamma']t_0)vv_1 \dots v_n && (\sigma_3) \\
  &\simeq_\sigma [\beta](\lambda x.(\mu\gamma[\gamma']t_0)v_1 \dots v_n)v && (\sigma_1)^n
  \end{aligned}$$
  soit il y a eu réduction multiplicative lors de la traduction de l'application aux $v_i$ et comme dans le lemme 11.24 on peut appliquer le lemme 11.23 à $[\gamma']u'$ d'où $[\gamma']u' = N[[\gamma]u'']$ et :
  $$\begin{aligned}
  t &= [\beta](\mu\gamma[\gamma']u')v_1 \dots v_n \\
  &\simeq_\sigma [\beta](\mu\gamma.N[[\gamma]u''])v_1 \dots v_n \\
  &\simeq_\sigma N[[\beta](u'')v_1 \dots v_n] && \text{lemme 11.20} \\
  &\simeq_\sigma N[[\beta](\lambda x.t_0)v] && \text{récurrence} \\
  &\simeq_\sigma [\beta](\mu\alpha'.N[[\alpha']\lambda x.t_0])v && \text{lemme 11.20} \\
  &\simeq_\sigma [\beta](\mu\alpha'[\alpha']\lambda x.\mu\alpha'.N[[\alpha']t_0])v && \text{lemme 11.19} \\
  &\simeq_\sigma [\beta](\lambda x.\mu\alpha'.N[[\alpha']t_0])v && (\theta)
  \end{aligned}$$

□

11.27. Théorème (Complétude)
*Soient $t$ et $t'$ deux $\lambda\mu$-termes, $t^\sim \simeq_s t'^\sim \Rightarrow t \simeq_\sigma t'$.*

Démonstration : On raisonne par récurrence sur la taille de ce réseau $\mathcal{R} = t^\sim$ en montrant simultanément le résultat pour les termes et les termes nommés et en suivant le même schéma que dans la preuve du théorème 5.19. Si $t$ est un terme nommé, $\mathcal{R}$ n'a pas de conclusion distinguée et on considère les cas suivants :

- Si $\mathcal{R}$ a un nœud ⅋ terminal dont la conclusion correspond à la variable $\alpha$, par le lemme 11.24, il existe des termes $u$ et $u'$ tels que $t \simeq_\sigma [\alpha]\lambda x.u$ et $t' \simeq_\sigma [\alpha]\lambda x.u'$ avec $u^\sim \simeq_s u'^\sim$. On a alors, par hypothèse de récurrence, $u \simeq_\sigma u'$ et donc $t \simeq_\sigma t'$.
- Si $\mathcal{R}$ a un nœud $?c$ terminal de type $?P$ dont la conclusion correspond à la variable $x$, il existe $u$ et $u'$ tels que $t = u[^x/_y]$ et $t' = u'[^x/_y]$ et $u^\sim = u'^\sim$ est le réseau obtenu en supprimant ce nœud $?c$ dans $\mathcal{R}$. Par hypothèse de récurrence, $u \simeq_\sigma u'$ et donc $t \simeq_\sigma t'$.

– Si $\mathcal{R}$ a un nœud $?c$ terminal de type $N \neq ?P$, idem.
– Si $\mathcal{R}$ a un nœud $?w$ terminal de type $N \neq ?P$, idem.
– Si $\mathcal{R}$ n'a pas de nœuds négatifs terminaux mais contient une coupure maximale dont l'arbre positif n'est pas plat, par le lemme 11.25, il existe $u$, $u'$, $v$ et $v'$ tels que $t \simeq_\sigma [\alpha](u)v$ et $t' \simeq_\sigma [\alpha](u')v'$, or par hypothèse de récurrence $u \simeq_\sigma u'$ et $v \simeq_\sigma v'$ donc $t \simeq_\sigma t'$.
– Si $\mathcal{R}$ n'a pas de nœuds négatifs terminaux mais contient une coupure maximale dont l'arbre positif est plat, par le lemme 11.26, il existe $u$, $u'$, $v$ et $v'$ tels que $t \simeq_\sigma [\alpha](\lambda x.u)v$ et $t' \simeq_\sigma [\alpha](\lambda x.u')v'$. Par hypothèse de récurrence, $[\alpha]u \simeq_\sigma [\alpha]u'$ et $v \simeq_\sigma v'$ donc $t \simeq_\sigma t'$.
– Si $\mathcal{R}$ n'a ni nœud négatif terminal ni coupure à profondeur 0 alors $\mathcal{R}$ est un arbre positif (avec un nœud $?d$ au-dessous), $t = (x)v_1 \dots v_n$ et $t' = (x)v'_1 \dots v'_n$, on peut appliquer l'hypothèse de récurrence aux $v_i$ ce qui donne pour tout $i$, $v_i \simeq_\sigma v'_i$ et donc $t \simeq_\sigma t'$.

Si $t$ et $t'$ sont des termes (non nommés), soit $\alpha$ une variable n'apparaissant ni dans $t$ ni dans $t'$, on a $t \simeq_\sigma \mu\alpha[\alpha]t$ et $t' \simeq_\sigma \mu\alpha[\alpha]t'$ avec $([\alpha]t)^\sim \simeq_s ([\alpha]t')^\sim$ de même taille que $t^\sim$, on peut donc appliquer le cas des termes nommés ce qui donne $[\alpha]t \simeq_\sigma [\alpha]t'$ et donc $t \simeq_\sigma t'$. □

### 11.4.3 Propriétés

11.28. PROPOSITION
*Soient $u$ et $v$ deux $\lambda\mu$-termes, si $u \simeq_\sigma v$ alors $u \simeq_{\beta\eta\mu\rho\theta} v$.*

Démonstration : On va montrer que l'on peut démontrer chacune des équations de la $\sigma$-équivalence à l'aide de celles de la $\beta\eta\mu\rho\theta$-équivalence :

($\sigma_i$) Réalisée par la $\beta$-équivalence.

(`push`/`push`) Réalisée par la $\mu\rho$-équivalence :

$$\begin{aligned}
[\alpha'](\mu\alpha[\beta'](\mu\beta.u)v)w &\simeq_\mu [\alpha'](\mu\alpha[\beta']\mu\beta.u[^{[\beta](t)v}/_{[\beta]t}])w \\
&\simeq_\rho [\alpha'](\mu\alpha.u[^{[\beta](t)v}/_{[\beta]t}][^{\beta'}/_{\beta}])w \\
&= ([\alpha'](\mu\alpha.u[^{[\beta](t)v}/_{[\beta]t}])w)[^{\beta'}/_{\beta}] && \beta \notin w \\
&= ([\alpha'](\mu\alpha.u)w)[^{[\beta](t)v}/_{[\beta]t}][^{\beta'}/_{\beta}] && \alpha \notin v \\
&\simeq_\rho [\beta']\mu\beta.([\alpha'](\mu\alpha.u)w[^{[\beta](t)v}/_{[\beta]t}]) \\
&\simeq_\mu [\beta'](\mu\beta[\alpha'](\mu\alpha.u)w)v
\end{aligned}$$

(`push`/`pop`) Réalisée de la même manière par la $\mu\rho$-équivalence.

(`pop`/`pop`) Il faut ici la $\beta\eta\mu\rho$-équivalence :

$$\begin{aligned}
[\alpha']\lambda x.\mu\alpha[\beta']\lambda y.\mu\beta.u &\simeq_\rho [\beta']\mu\beta[\alpha']\lambda x.\mu\alpha[\beta]\lambda y.\mu\beta.u && \beta \neq \alpha \text{ et } \beta \neq \alpha' \\
&\simeq_\eta [\beta']\lambda y.(\mu\beta[\alpha']\lambda x.\mu\alpha[\beta]\lambda y.\mu\beta.u)y && y \neq x \\
&\simeq_\mu [\beta']\lambda y.\mu\beta[\alpha']\lambda x.\mu\alpha[\beta](\lambda y.\mu\beta.u)y \\
&\simeq_\beta [\beta']\lambda y.\mu\beta[\alpha']\lambda x.\mu\alpha[\beta]\mu\beta.u \\
&\simeq_\rho [\beta']\lambda y.\mu\beta[\alpha']\lambda x.\mu\alpha.u
\end{aligned}$$

($\rho$) Réalisée par la $\rho$-équivalence.

($\theta$) Réalisée par la $\theta$-équivalence. □

La $\sigma$-équivalence réalise des identifications entre termes qui sont plus complexes que dans le cas du $\lambda$-calcul. En particulier elle permet d'identifier des termes normaux et non normaux (y compris si on considère les termes normaux pour les $\rho$- et $\theta$-réductions) : soit $t$ un terme clos et $u = \lambda x.\mu\beta[\beta](\mu\alpha[\beta]\lambda y.\mu\delta[\alpha]x)t$ et $v = \lambda x.\mu\beta[\beta]\lambda y.\mu\delta[\beta](x)t$, on a $u \simeq_\sigma v$ (c'est une variante de (`push`/`pop`) et un cas particulier du lemme 11.20) et on a également $u \to^+ v$ :

$$u \quad \to_\mu \quad \lambda x.\mu\beta[\beta]\mu\alpha[\beta]\lambda y.\mu\delta[\alpha](x)t \quad \to_\rho \quad v$$

Il n'y a donc aucun espoir que la $\sigma$-équivalence préserve les longueurs de réduction comme c'est le cas pour le $\lambda$-calcul [Reg92]. La raison est que le $\lambda\mu$-calcul contient des *$\mu$-rédex linéaires* qui n'ont pas de réelle signification opérationnelle. Plus précisément, nous allons maintenant montrer que la $\sigma$-équivalence identifie les termes qui ne différent que par des $\mu$-rédex linéaires.

11.29. Définition ($\mu$-rédex linéaire)
Un $\mu$-rédex est dit *linéaire* s'il est de la forme $(\mu\alpha.N[[\alpha]u])v$ où $N$ est un contexte nommé sans $\alpha$ et $\alpha \notin u$. La *taille* du $\mu$-rédex est le nombre de $\mu$ dans le terme $\mu\alpha.N[[\alpha]u]$ après avoir remplacé toute application $(t)t'$ par $t$.

On montre que la $\mu$-réduction linéaire termine et est incluse dans la $\sigma$-équivalence.

11.30. Proposition (Élimination des $\mu$-rédex linéaires)
*Soit $u$ un $\lambda\mu$-terme, la $\mu$-réduction linéaire de $u$ termine et on obtient un terme $u'$ tel que $u' \simeq_\sigma u$.*

Démonstration : On montre que tout $\mu$-rédex linéaire peut être soit remplacé par un plus petit soit éliminé et de manière compatible avec la $\sigma$-équivalence :

$$\begin{aligned}(\mu\alpha.N[[\alpha]u])v &\simeq_\sigma \mu\beta[\beta](\mu\alpha.N[[\alpha]u])v && (\theta)\\ &\simeq_\sigma \mu\beta.N[[\beta](u)v] && \text{lemme 11.20}\end{aligned}$$

donc soit le nombre de $\mu$-rédex linéaires diminue soit la taille d'un de ces rédex, au moins, diminue sans que celle des autres augmente. □

11.31. Lemme (Réduction infinie)
*Si $u$ admet une suite infinie de réductions, alors $u^\sim$ également.*

Démonstration : Certaines $\mu$-réductions (en particulier les $\mu$-réductions linéaires) sont traduites par 0 étapes de réduction dans les réseaux. Il suffit de remarquer que la réciproque est vraie : si $u \to_\mu v$ et $u^\sim = v^\sim$ alors la $\mu$-réduction correspond nécessairement à un $\mu$-rédex linéaire, or il ne peut pas y avoir de suite infinie de $\mu$-réductions linéaires dans un $\lambda\mu$-terme. On en déduit que toute réduction infinie d'un $\lambda\mu$-terme contient un nombre infini de réductions qui ne soient pas des $\mu$-réductions linéaires, donc $u^\sim$ a une suite infinie de réductions. □

11.32. Proposition (Propriétés de préservation)
*Soient $u$ et $v$ deux $\lambda\mu$-termes tels que $u \simeq_\sigma v$.*
- *Si $u$ est normalisable alors $v$ est normalisable et leurs formes normales sont $\sigma$-équivalentes.*
- *Si $u$ admet une forme normale de tête alors $v$ aussi.*
- *Si $u$ est fortement normalisable, alors $v$ est fortement normalisable.*
- *Si $u$ est typable de type $A$ alors $v$ est typable de type $A$.*

Démonstration : Il s'agit essentiellement de corollaires du théorème 11.27 qui pourraient également être démontrés directement à l'aide des équations de la $\sigma$-équivalence.
- Si $u$ est normalisable alors $u^\sim$ est normalisable par simulation et puisque $v^\sim \simeq_s u^\sim$, $v$ est normalisable par la proposition 11.15.
- De la même manière que le cas précédent en remarquant que la réduction de tête du $\lambda\mu$-calcul correspond à la réduction à profondeur 0 des réseaux.
- Grâce au lemme 11.31, et en raisonnant comme dans la proposition 11.15.
- Si $u$ est typable de type $A$ alors $u^\sim \simeq_s v^\sim$ est typable de type $A^-$ (définition 11.2) et en suivant les étapes de la traduction de $v$ ceci donne une dérivation de typage de $v$ avec le type $A$. □

11.33. Exemple
Regardons les transformations réalisées sur les dérivations de typage par la $\sigma$-équivalence. Par exemple avec l'équation (`push`/`pop`), si $\Gamma, x : A \vdash u \mid \alpha : C \to D, \beta : B, \Delta$ et $\Gamma' \vdash v : C \mid \Delta'$, on a :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, x : A \vdash u \mid \alpha : C \to D, \beta : B, \Delta}{\Gamma, x : A \vdash \mu\beta.u : B \mid \alpha : C \to D, \Delta}\,\mu-\text{abs}}{\Gamma \vdash \lambda x.\mu\beta.u : A \to B \mid \alpha : C \to D, \Delta}\,\text{abs}}{\Gamma \vdash \mu\alpha[\beta']\lambda x.\mu\beta.u : C \to D \mid \beta' : A \to B, \Delta}\,\mu \qquad \Gamma' \vdash v : C \mid \Delta'}{\Gamma, \Gamma' \vdash (\mu\alpha[\beta']\lambda x.\mu\beta.u)v : D \mid \beta' : A \to B, \Delta, \Delta'}\,\text{app}}{\Gamma, \Gamma' \vdash [\alpha'](\mu\alpha[\beta']\lambda x.\mu\beta.u)v \mid \alpha' : D, \beta' : A \to B, \Delta, \Delta'}\,\text{name}$$

et

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, x : A \vdash u \mid \alpha : C \to D, \beta : B, \Delta}{\Gamma, x : A \vdash \mu\alpha.u : C \to D \mid \beta : B, \Delta}\,\mu-\text{abs} \qquad \Gamma' \vdash v : C \mid \Delta'}{\Gamma, \Gamma', x : A \vdash (\mu\alpha.u)v : D \mid \beta : B, \Delta, \Delta'}\,\text{app}}{\Gamma, \Gamma', x : A \vdash \mu\beta[\alpha'](\mu\alpha.u)v : B \mid \alpha' : D, \Delta, \Delta'}\,\mu}{\Gamma, \Gamma' \vdash \lambda x.\mu\beta[\alpha'](\mu\alpha.u)v : A \to B \mid \alpha' : D, \Delta, \Delta'}\,\text{abs}}{\Gamma, \Gamma' \vdash [\beta']\lambda x.\mu\beta[\alpha'](\mu\alpha.u)v \mid \alpha' : D, \beta' : A \to B, \Delta, \Delta'}\,\text{name}$$

Depuis le début de ce chapitre, nous avons considéré le $\lambda\mu$-calcul tel qu'il a été introduit par M. Parigot et nous avons étudié en détails les liens avec les réseaux polarisés. Toutefois ceci n'utilise qu'un *fragment* de LLP, pas de connecteurs additifs en particulier. Toute cette étude peut être étendue au $\lambda\mu$-calcul complet (annexe D.5) et on montrerait que la traduction utilise alors l'intégralité des connecteurs de LLP, plus précisément on pourrait énoncer un résultat de surjectivité de la traduction des termes en réseaux.

Il faut cependant remarquer que les règles de réduction du $\lambda\mu$-calcul de Selinger (donc du $\lambda\mu$-calcul complet) ne sont pas définies (excepté par une CPS-traduction). En considérant que de telles règles devraient mener à une généralisation du théorème 11.9, on pourrait les définir à partir de la procédure d'élimination des coupures de LLP et elles seraient automatiquement compatibles avec la sémantique des catégories de contrôle. Donner explicitement toutes ces règles serait fastidieux et donnerait lieu à de nombreuses règles de réduction commutative quotientées par la (généralisation de la) $\sigma$-équivalence.

## 11.5 Autres $\lambda$-systèmes classiques par nom

### 11.5.1 $\lambda c$-calcul

Nous étudions ici la traduction du $\lambda c$-calcul [Kri94] (variante du calcul de Felleisen [Fel86]) dans LLP.

11.34. Définition ($\lambda c$-calcul)
Les termes du *$\lambda c$-calcul* sont donnés par :

$$t \quad ::= \quad x \quad \mid \quad \lambda x.t \quad \mid \quad (t)t \quad \mid \quad c$$

Les règles de réduction sont la $\beta$-réduction et :

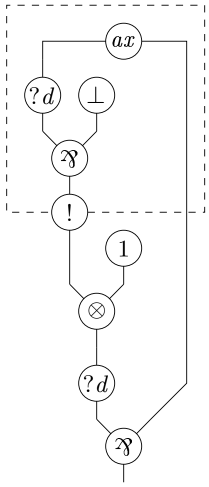
$$(c)tt_1 \dots t_n \quad \to_c \quad (t)\lambda x.(x)t_1 \dots t_n$$

Le système de typage associé à ce calcul est celui du $\lambda$-calcul enrichi par la règle :

$$\overline{\Gamma \vdash c : ((A \to \bot) \to \bot) \to A}\ c$$

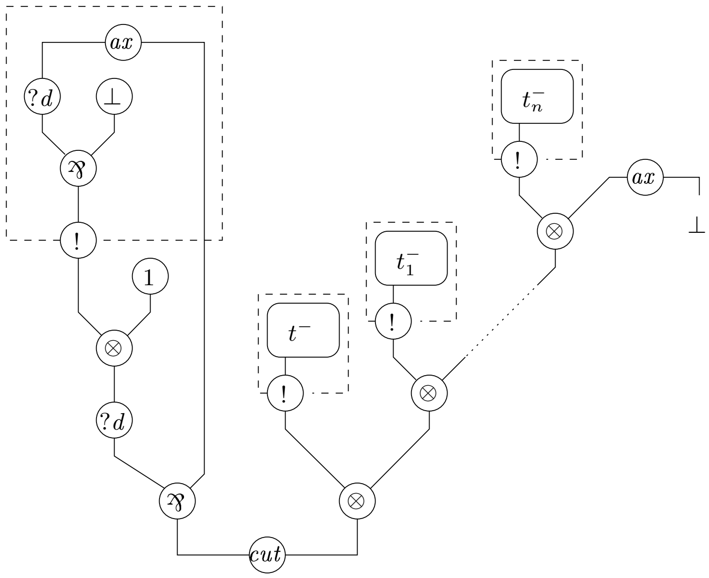
On ne considère la réduction $\to_c$ que si le terme $(c)tt_1 \ldots t_n$ a type $\bot$.

En utilisant la traduction négative $(.)^-$ pour le $\lambda$-calcul (comme décrite pour le $\lambda\mu$-calcul) et en traduisant la constante $c$ par le réseau :

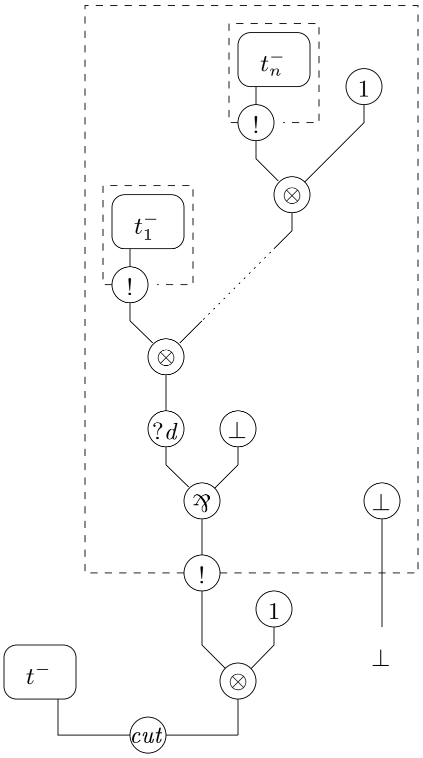



on obtient une simulation de la réduction du $\lambda c$-calcul par celle des réseaux polarisés modulo $\eta$-expansion sur les axiomes de type $\vdash 1, \bot$ et commutation des nœuds $\bot$ avec les portes auxiliaires de boîtes (ce qui correspond à la réversibilité de $\bot$ ou à l'équivalence structurelle pour les affaiblissements).

Démonstration : Grâce au théorème 11.9, il suffit de montrer la simulation de la réduction $\to_c$. Le réseau $((c)tt_1 \ldots t_n)^-$ :




donne, par réduction et $\eta$-expansion, le réseau :




qui est (modulo commutation de $\perp$ avec la boîte) une $\eta$-expansion de $((t)\lambda x.(x)t_1 \dots t_n)^-$. $\square$

### 11.5.2 $\lambda\kappa$-calcul

Nous étudions ici la traduction du $\lambda\kappa$-calcul [DK00] dans LLP.

11.35. Définition ($\lambda\kappa$-calcul)
Les termes du *$\lambda\kappa$-calcul* sont donnés avec les piles de termes par :

$$\begin{array}{rcl} t & ::= & x \mid \lambda x.t \mid (t)t \mid \kappa x.t \mid \star_t \mid \star_\pi \\ \pi & ::= & \varepsilon \mid t.\pi \end{array}$$

Un programme est une paire $t, \pi$. Les règles de réduction pour les programmes sont :

$$\begin{array}{rcl} (t)u, \pi & \to & t, u.\pi \\ \lambda x.t, u.\pi & \to & t[{}^{\star_u}/_x], \pi \\ \star_u, \pi & \to & u, \pi \\ \kappa x.t, \pi & \to & t[{}^{\star_\pi}/_x], \pi \\ \star_\pi, t.\pi' & \to & t, \pi \end{array}$$

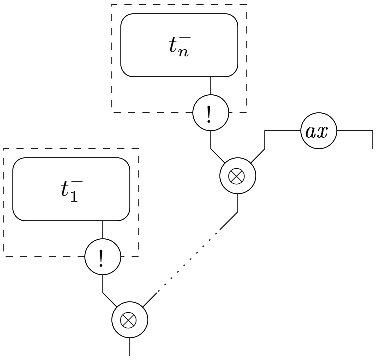
Le système de typage des termes associé à ce calcul est celui du $\lambda$-calcul enrichi par la règle :

$$\frac{\Gamma, x : A \to B \vdash t : A}{\Gamma \vdash \kappa x.t : A}\,\kappa$$

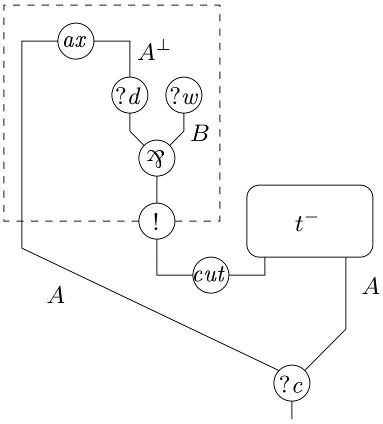
Concernant les programmes, si $t : A_1 \to \cdots \to A_n \to A$ et si les termes de $\pi$ ont types $A_1$, ..., $A_n$, alors $t, \pi$ a type $A$.

On utilise la traduction négative $(.)^-$ pour le $\lambda$-calcul, en ajoutant pour les nouvelles constructions :

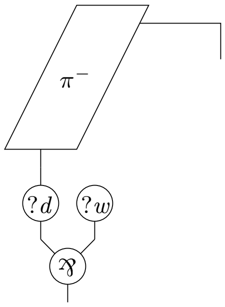
$\pi = t_1 \ldots t_n$




$\kappa x.t$

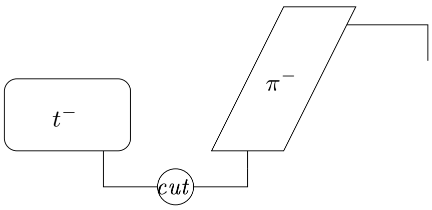



$\star_\pi$




et la traduction de $\star_t$ est la même que celle de $t$.

Le programme $t, \pi$ est traduit en plaçant une coupure entre la traduction $t$ et celle de $\pi$ :

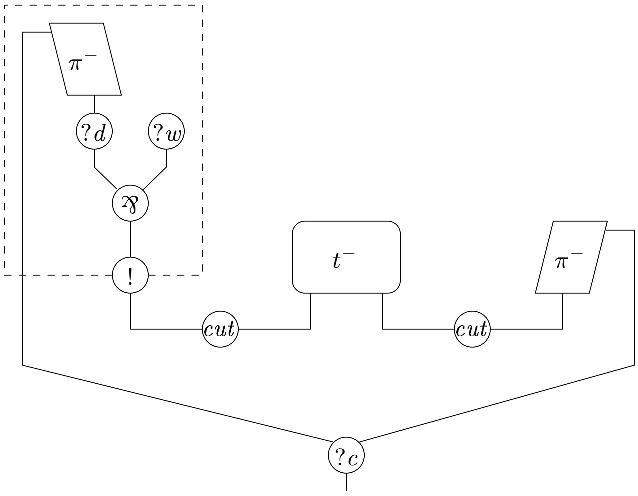



11.36. Proposition

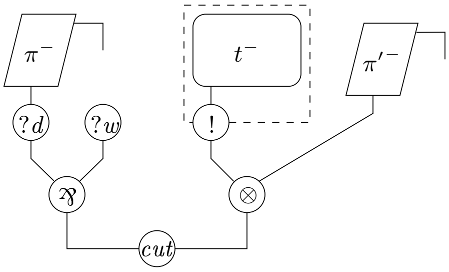
*La réduction des réseaux polarisés simule celle du $\lambda\kappa$-calcul par la traduction ci-dessus.*

Démonstration : On considère chacune des règles de réduction :

– $(t)u, \pi \to t, u.\pi$ : cette règle est simulée par une simple réduction $ax$.
– $\lambda x.t, u.\pi \to t[^{\star_u}/_x], \pi$ : la traduction du programme $\lambda x.t, u.\pi$ est la même que celle du terme $(\lambda x.t)ut_1 \dots t_n$ où $\pi = t_1 \dots t_n$, or d'après le théorème 11.9 le réseau obtenu se réduit en la traduction de $(t[^u/_x])t_1 \dots t_n$ qui se réduit en celle du programme $t[^{\star_u}/_x], \pi$ par des réductions $ax$.
– $\star_u, \pi \to u, \pi$ : les deux programmes sont traduits à l'identique.
– $\kappa x.t, \pi \to t[^{\star_\pi}/_x], \pi$ : par réduction de $(\kappa x.t, \pi)^-$, on obtient le réseau :




qui, d'après le lemme 11.7, se réduit en $(t[^{\star_\pi}/_x], \pi)^-$.
– $\star_\pi, t.\pi' \to t, \pi$ : le réseau $(\star_\pi, t.\pi')^-$ :




se réduit en $(t, \pi)^-$ modulo un nœud $?w$ terminal. □

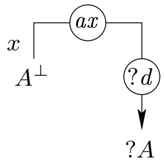
11.37. Remarque : Le terme de l'exemple 11.5 s'écrit simplement en $\lambda\kappa$-calcul : $(\kappa x.x)\kappa x.x$.

## 11.6 Traduction positive

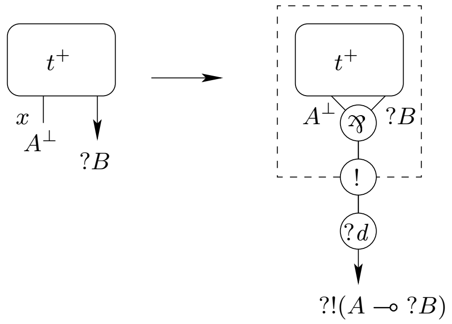
Comme on a traduit le $\lambda\mu$-calcul par nom dans LLP par la traduction négative, on va désormais traduire le $\lambda\mu$-calcul par valeur dans LLP par la *traduction positive*. Cette idée est initialement due à P. Selinger.

La traduction des différentes constructions des $\lambda\mu$-termes est donnée de la manière suivante :

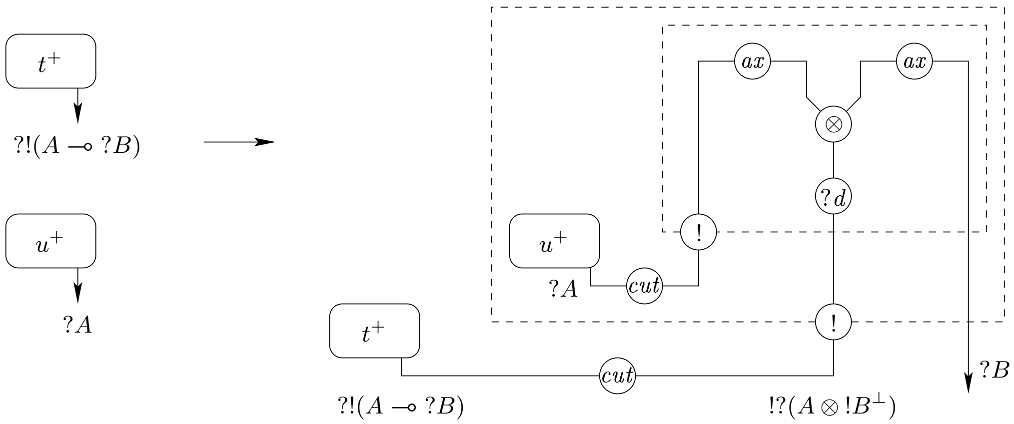
$(x)$



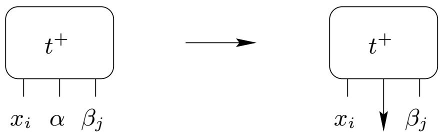
($\lambda$)



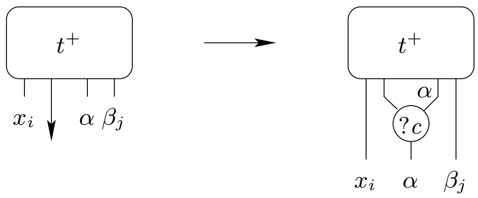

(@)




($\mu$)

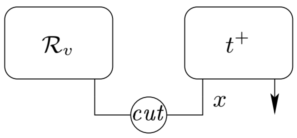



($\alpha$)




On obtient $t^+$ en éliminant les coupures $!/?d$ dans le réseau ainsi obtenu.

11.38. Lemme ($\lambda$-substitution)
*Soient $t$ et $v$ deux $\lambda\mu$-termes, si $v$ est une* valeur *et si on note $\mathcal{R}_v$ le réseau obtenu en supprimant le nœud $?d$ situé au-dessus de la conclusion distinguée de $v^+$, alors le réseau :*

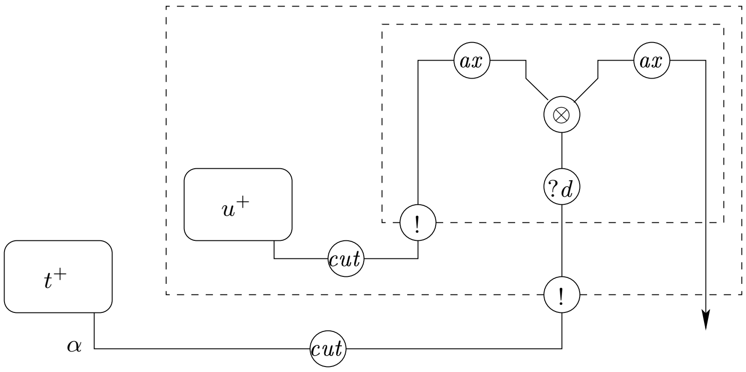



*se réduit en $(t[^v/_x])^+$.*

Démonstration : On raisonne par récurrence sur la taille de $t$ :

- Si $t = x$, le résultat est obtenu par une simple réduction $ax$.
- Si $t = \lambda x.u$, on applique l'hypothèse de récurrence à $u$.
- Si $t = (u)u'$, on applique l'hypothèse de récurrence à $u$ et à $u'$ s'ils contiennent tous les deux la variable $x$ et à celui qui la contient sinon.
- Si $t = \mu\beta[\alpha]u$, on applique l'hypothèse de récurrence à $u$. □

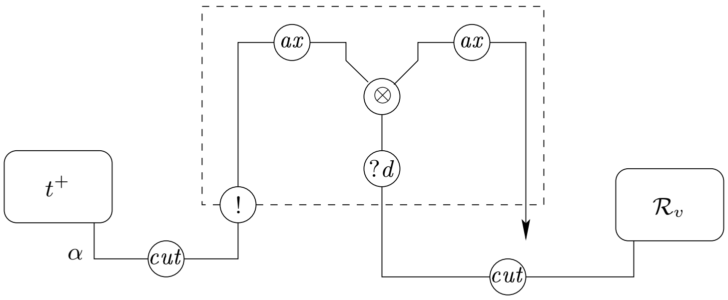
11.39. Lemme ($\mu$-substitution à droite)
*Soient $t$ et $u$ deux $\lambda\mu$-termes, le réseau :*




*se réduit en* $(t[^{[\alpha](w)u}/_{[\alpha]w}])^+$.

Démonstration : Comme pour le lemme précédent par récurrence sur la taille de $t$. □

11.40. Lemme ($\mu$-substitution à gauche)
*Soient $t$ et $v$ deux $\lambda\mu$-termes, si $v$ est une* valeur *et si on note $\mathcal{R}_v$ le réseau obtenu en supprimant le nœud $?d$ situé au-dessus de la conclusion distinguée de $v^+$, alors le réseau :*




*se réduit en* $(t[^{[\alpha](v)w}/_{[\alpha]w}])^+$.

Démonstration : Par récurrence sur la taille de $t$ :

- $t$ ne peut pas être une variable.
- Si $t = \lambda x.u$, on applique l'hypothèse de récurrence à $u$.
- Si $t = (u)u'$ et si un seul des deux termes contient la variable $\alpha$, il suffit d'appliquer l'hypothèse de récurrence à ce terme. Si $u$ et $u'$ contiennent $\alpha$, après deux étapes de réduction $!/?c$ (la première dupliquant la boîte et la seconde dupliquant $\mathcal{R}_v$), on applique l'hypothèse de récurrence à $u$ et à $u'$.
- Si $t = \mu\gamma[\beta]u$, soit $\beta \neq \alpha$ et on applique l'hypothèse de récurrence à $u$, soit $\beta = \alpha$ et $\alpha \in u$ et on applique deux étapes de réduction $!/?c$ avant d'appliquer l'hypothèse de récurrence à $u$, soit, enfin, $\beta = \alpha$ et $\alpha \notin u$ et le résultat est immédiat. □

11.41. PROPOSITION (SIMULATION)
*Si $u$ se réduit en $v$ (en appel par valeur) alors $u^+$ se réduit en $v^+$.*

Démonstration : On considère les différentes réductions possibles :

- ($\beta_v$) Après une réduction $!/!$, une réduction $!/?d$ et une réduction $\otimes/\parr$, le réseau $((\lambda x.t)v)^+$ donne le réseau du lemme 11.38 qui se réduit donc en $(t[^v/_x])^+$.
- ($\mu$) On applique directement le lemme 11.39.
- ($\mu_v$) Puisque les coupures $!/?d$ sont réduites dans $(.)^+$, on peut appliquer le lemme 11.40. □

Nous arrêtons ici la liste des $\lambda$-systèmes que l'on traduit dans LLP, il paraît naturel que d'autres extensions classiques déterministes du $\lambda$-calcul puissent également se traduire dans LLP.

# Chapitre 12

# Systèmes à variantes $\mathsf{Q}$ et $\mathsf{T}$

## 12.1 $\mathsf{LK}^{\mathsf{tq}}$

Le système $\mathsf{LK}^{\mathsf{tq}}$ est un système dont l'élimination des coupures est non déterministe, qui a été introduit par Danos-Joinet-Schellinx [DJS95] dans le cadre de leur analyse de la logique classique par la logique linéaire. Il possède deux sous-systèmes déterministes duaux $\mathsf{LKQ}$ et $\mathsf{LKT}$. Ces deux systèmes sont basés sur deux traductions différentes de l'implication classique en logique linéaire. Nous allons voir que ces traductions peuvent se factoriser par les traductions $(.)^+$ et $(.)^-$ dans $\mathsf{LLP}$.

### 12.1.1 $\mathsf{LKQ}$

Le système $\mathsf{LKQ}$ a été défini en utilisant la $\mathsf{q}$-traduction de la flèche : $!(A \multimap ?B)$. Nous allons voir qu'elle correspond à la traduction $(.)^+$ et permet de traduire $\mathsf{LKQ}$ dans $\mathsf{LLP}$.

$$\frac{}{A \vdash ; A}\,ax \qquad \frac{\Gamma \vdash \Delta ; A \qquad \Gamma', A \vdash \Delta' ; \Pi}{\Gamma, \Gamma' \vdash \Delta, \Delta' ; \Pi}\,cut \qquad \frac{\Gamma \vdash \Delta, A ; \Pi \qquad \Gamma', A \vdash \Delta' ;}{\Gamma, \Gamma' \vdash \Delta, \Delta' ; \Pi}\,cut$$

$$\frac{\Gamma, A \vdash \Delta, B ;}{\Gamma \vdash \Delta ; A \to B}\,\to_R \qquad \frac{\Gamma \vdash \Delta ; A \qquad \Gamma', B \vdash \Delta' ;}{\Gamma, \Gamma', A \to B \vdash \Delta, \Delta' ;}\,\to_L$$

$$\frac{\Gamma \vdash \Delta, A ;}{\Gamma \vdash \Delta ; \forall X A}\,\forall_R \qquad \frac{\Gamma, A[{}^{B}/_{X}] \vdash \Delta ;}{\Gamma, \forall X A \vdash \Delta ;}\,\forall_L$$

$$\frac{\Gamma \vdash \Delta ; A}{\Gamma \vdash \Delta, A ;}\,de$$

$$\frac{\Gamma \vdash \Delta ; \Pi}{\Gamma \vdash \Delta, A ; \Pi}\,wk_R \qquad \frac{\Gamma \vdash \Delta ; \Pi}{\Gamma, A \vdash \Delta ; \Pi}\,wk_L \qquad \frac{\Gamma \vdash \Delta, A, A ; \Pi}{\Gamma \vdash \Delta, A ; \Pi}\,co_R \qquad \frac{\Gamma, A, A \vdash \Delta ; \Pi}{\Gamma, A \vdash \Delta ; \Pi}\,co_L$$

On traduit $\mathsf{LKQ}$ dans $\mathsf{LLP}$ par la traduction $(.)^+$ en traduisant les séquents par :

$$\Gamma \vdash \Delta ; \Pi \qquad \rightsquigarrow \quad \vdash \Gamma^{+\perp}, ?\Delta^+, \Pi^+$$

Cette traduction arrive en fait dans $\mathsf{LL}_{\mathsf{pol}}$, les règles structurelles généralisées ne sont pas nécessaires mais ceci est uniquement le cas parce qu'on n'utilise que les connecteurs $\to$ et $\forall$, des connecteurs comme $\wedge$ ou $\exists$ nécessiteraient de considérer $\mathsf{LLP}$ à la place de $\mathsf{LL}_{\mathsf{pol}}$.

12.1. PROPOSITION (SIMULATION)
*Par la traduction* $(.)^+$*, la réduction de* $\mathsf{LL}_{\mathrm{pol}}$ *simule celle de* $\mathsf{LKQ}$.

Démonstration : Puisqu'on ne considère que le *fragment* $\rightarrow, \forall$ ce résultat est celui pour la $\mathtt{q}$-traduction de [DJS95]. □

### 12.1.2 LKT

Le système LKT a été défini en utilisant la $\mathtt{t}$-traduction de la flèche : $!?A \multimap ?B$. Nous allons voir qu'elle correspond à la traduction $(.)^-$ et permet de traduire LKT dans LLP.

$$\frac{}{A; \vdash A}\,ax \qquad \frac{\Pi; \Gamma \vdash \Delta, A \qquad A; \Gamma' \vdash \Delta'}{\Pi; \Gamma, \Gamma' \vdash \Delta, \Delta'}\,cut \qquad \frac{; \Gamma \vdash \Delta, A \qquad \Pi; \Gamma', A \vdash \Delta'}{\Pi; \Gamma, \Gamma' \vdash \Delta, \Delta'}\,cut$$

$$\frac{\Pi; \Gamma, A \vdash \Delta, B}{\Pi; \Gamma \vdash \Delta, A \rightarrow B}\rightarrow_R \qquad \frac{; \Gamma \vdash \Delta, A \qquad B; \Gamma' \vdash \Delta'}{A \rightarrow B; \Gamma, \Gamma' \vdash \Delta, \Delta'}\rightarrow_L$$

$$\frac{\Pi; \Gamma \vdash \Delta, A}{\Pi; \Gamma \vdash \Delta, \forall X A}\forall_R \qquad \frac{A[^B/_X]; \Gamma \vdash \Delta;}{\forall X A; \Gamma \vdash \Delta;}\forall_L$$

$$\frac{A; \Gamma \vdash \Delta}{; A, \Gamma \vdash \Delta}\,de$$

$$\frac{\Pi; \Gamma \vdash \Delta}{\Pi; \Gamma \vdash \Delta, A}\,wk_R \qquad \frac{\Pi; \Gamma \vdash \Delta}{\Pi; \Gamma, A \vdash \Delta}\,wk_L \qquad \frac{\Pi; \Gamma \vdash \Delta, A, A}{\Pi; \Gamma \vdash \Delta, A}\,co_R \qquad \frac{\Pi; \Gamma, A, A \vdash \Delta}{\Pi; \Gamma, A \vdash \Delta}\,co_L$$

On traduit LKT dans LLP par la traduction $(.)^-$ en traduisant les séquents par :

$$\Pi; \Gamma \vdash \Delta \qquad \rightsquigarrow \quad \vdash \Pi^{-\perp}, ?\Gamma^{-\perp}, \Delta^-$$

En composant cette traduction avec la traduction $(.)^b$, on retrouve la $\mathtt{t}$-traduction de LKT dans LL :

$$A \rightarrow B \qquad \rightsquigarrow \quad !A \multimap B \qquad \rightsquigarrow \quad !?A \multimap ?B$$

Cette remarque sera appliquée au $\lambda\mu$-calcul dans la section 13.1.

12.2. PROPOSITION (SIMULATION)
*Par la traduction* $(.)^-$*, la réduction de* LLP *simule celle de* LKT.

Démonstration : Il s'agit essentiellement d'un cas particulier de la proposition 10.6. □

## 12.2 $\bar{\lambda}\mu\tilde{\mu}$-calcul

Le $\bar{\lambda}\mu\tilde{\mu}$-calcul, introduit par Curien-Herbelin [CH00], est un calcul non déterministe pour la logique classique. Il possède deux sous-systèmes déterministes ($\bar{\lambda}\mu\tilde{\mu}_Q$-calcul et $\bar{\lambda}\mu\tilde{\mu}_T$-calcul) basés sur les deux orientations de la paire critique :

$$\begin{array}{lll} (\mu\beta.c, e) & \rightarrow_\mu & c[^e/_\beta] \\ (v, \tilde{\mu}x.c) & \rightarrow_{\tilde{\mu}} & c[^v/_x] \end{array}$$

qui codent respectivement l'appel par valeur et l'appel par nom.

### 12.2.1 $\bar{\lambda}\mu\tilde{\mu}_Q$-calcul

Les termes du $\bar{\lambda}\mu\tilde{\mu}_Q$-calcul sont principalement divisés en trois catégories : les commandes (ou programmes) $c$, les termes $v$ et $V$ et les environnements (ou continuations) $e$.

$$
\begin{array}{lcccccccc}
c & ::= & (v,e) & & & & & & \\
V & ::= & x & | & \lambda x.v & | & e.V & & \\
v & ::= & \mu\beta.c & | & V & & & & \\
e & ::= & \alpha & | & \tilde{\mu}x.c & | & V.e & | & \beta\lambda.e
\end{array}
$$

Les règles de réduction obtenues en déterminisant la paire critique du $\bar{\lambda}\mu\tilde{\mu}$-calcul par la distinction entre $v$ et $V$ (ce qui revient à d'abord effectuer les réductions $\mu$) sont :

$$
\begin{array}{rcl}
(\lambda x.v_1, V_2.e) & \to & (V_2, \tilde{\mu}x.(v_1,e)) \\
(e_2.V, \beta\lambda.e_1) & \to & (\mu\beta.(V,e_1), e_2) \\
(\mu\beta.c, e) & \to & c[{}^{e}/_{\beta}] \\
(V, \tilde{\mu}x.c) & \to & c[{}^{V}/_{x}]
\end{array}
$$

Les règles de typage de ce calcul sont proches des règles du système logique $\mathsf{LKQ}$, avec quatre sortes de jugements (une par catégorie syntaxique) :

$$
\begin{array}{c}
c : (\Gamma \vdash \Delta) \\
\Gamma \vdash V : A ; \Delta \\
\Gamma \vdash v : A \mid \Delta \\
\Gamma \mid e : A \vdash \Delta
\end{array}
$$

Ce qui donne, en suivant la définition des termes :

$$
\frac{}{x : A \vdash x : A;}\,ax \qquad \frac{}{\mid \alpha : A \vdash \alpha : A}\,ax \qquad \frac{\Gamma \vdash v : A \mid \Delta \qquad \Gamma' \mid e : A \vdash \Delta'}{(v,e) : (\Gamma, \Gamma' \vdash \Delta, \Delta')}\,cut
$$

$$
\frac{\Gamma, x : A \vdash v : B \mid \Delta}{\Gamma \vdash \lambda x.v : A \to B; \Delta}\to_R \qquad \frac{\Gamma \vdash V : A; \Delta \qquad \Gamma' \mid e : B \vdash \Delta'}{\Gamma, \Gamma' \mid V.e : A \to B \vdash \Delta, \Delta'}\to_L
$$

$$
\frac{\Gamma \mid e : A \vdash \Delta \qquad \Gamma' \vdash V : B; \Delta'}{\Gamma, \Gamma' \vdash e.V : B - A; \Delta, \Delta'}-_R \qquad \frac{\Gamma \mid e : B \vdash \beta : A, \Delta}{\Gamma \mid \beta\lambda.e : B - A \vdash \Delta}-_L
$$

$$
\frac{\Gamma \vdash V : A; \Delta}{\Gamma \vdash V : A \mid \Delta}\,de \qquad \frac{c : (\Gamma, x : A \vdash \Delta)}{\Gamma \mid \tilde{\mu}x.c : A \vdash \Delta}\,\tilde{\mu} \qquad \frac{c : (\Gamma \vdash \beta : B, \Delta)}{\Gamma \vdash \mu\beta.c : B \mid \Delta}\,\mu
$$

$$
\frac{\Gamma \mid \Pi \vdash \Pi \mid \Delta}{\Gamma \mid \Pi \vdash \Pi \mid \Delta, A}\,wk_R \qquad \frac{\Gamma \mid \Pi \vdash \Pi \mid \Delta}{\Gamma, A \mid \Pi \vdash \Pi \mid \Delta}\,wk_L
$$

$$
\frac{\Gamma \mid \Pi \vdash \Pi \mid \Delta, A, A}{\Gamma \mid \Pi \vdash \Pi \mid \Delta, A}\,co_R \qquad \frac{\Gamma, A, A \mid \Pi \vdash \Pi \mid \Delta}{\Gamma, A \mid \Pi \vdash \Pi \mid \Delta}\,co_L
$$

Ce qui signifie que l'on a les règles structurelles sur les contextes pour tous les types de jugements.

On traduit le $\bar{\lambda}\mu\tilde{\mu}_Q$-calcul dans LLP par la traduction $(.)^+$ en traduisant les jugements par :

$$\begin{array}{ccc}
c : (\Gamma \vdash \Delta) & \rightsquigarrow & \vdash \Gamma^{+\perp}, ?\Delta^+ \\
\Gamma \vdash V : A; \Delta & \rightsquigarrow & \vdash \Gamma^{+\perp}, ?\Delta^+, A^+ \\
\Gamma \vdash v : A \mid \Delta & \rightsquigarrow & \vdash \Gamma^{+\perp}, ?\Delta^+, ?A^+ \\
\Gamma \mid e : A \vdash \Delta & \rightsquigarrow & \vdash A^{+\perp}, \Gamma^{+\perp}, ?\Delta^+
\end{array}$$

En composant cette traduction avec la traduction $(.)^b$, on retrouve la traduction du $\bar{\lambda}\mu\tilde{\mu}_Q$-calcul dans LL décrite par Curien-Herbelin.

12.3. Proposition (Simulation)
*Par la traduction $(.)^+$, la réduction de* LLP *simule celle du $\bar{\lambda}\mu\tilde{\mu}_Q$-calcul.*

Démonstration : Dans la mesure où la réduction du $\bar{\lambda}\mu\tilde{\mu}_Q$-calcul correspond à l'élimination des coupures de LKQ, il s'agit essentiellement d'un corollaire de la proposition 12.1. □

### 12.2.2 $\bar{\lambda}\mu\tilde{\mu}_T$-calcul

Les termes du $\bar{\lambda}\mu\tilde{\mu}_T$-calcul possèdent les mêmes catégories syntaxiques que pour ceux du $\bar{\lambda}\mu\tilde{\mu}_Q$-calcul avec une distinction entre deux types d'environnements $e$ et $E$ cette fois.

$$\begin{array}{rcccccccc}
c & ::= & (v, e) \\
v & ::= & x & \mid & \mu\beta.c & \mid & \lambda x.v & \mid & E.v \\
E & ::= & \alpha & \mid & v.E & \mid & \beta\lambda.e \\
e & ::= & \tilde{\mu}x.c & \mid & E
\end{array}$$

Les règles de réduction obtenues en déterminisant la paire critique du $\bar{\lambda}\mu\tilde{\mu}$-calcul par la distinction entre $e$ et $E$ (ce qui revient à d'abord effectuer les réductions $\tilde{\mu}$) sont :

$$\begin{array}{rcl}
(\lambda x.v_1, v_2.E) & \rightarrow & (v_2, \tilde{\mu}x.(v_1, E)) \\
(E_2.v, \beta\lambda.e_1) & \rightarrow & (\mu\beta.(v, e_1), E_2) \\
(\mu\beta.c, E) & \rightarrow & c[{}^{E}/_{\beta}] \\
(v, \tilde{\mu}x.c) & \rightarrow & c[{}^{v}/_{x}]
\end{array}$$

Les règles de typage de ce calcul sont, cette fois, proches des règles du système logique LKT, avec toujours quatre sortes de jugements :

$$\begin{array}{c}
c : (\Gamma \vdash \Delta) \\
\Gamma \vdash v : A \mid \Delta \\
\Gamma; E : A \vdash \Delta \\
\Gamma \mid e : A \vdash \Delta
\end{array}$$

Ce qui donne :

$$\frac{}{x : A \vdash x : A \mid}\, ax \qquad \frac{}{; \alpha : A \vdash \alpha : A}\, ax \qquad \frac{\Gamma \vdash v : A \mid \Delta \qquad \Gamma' \mid e : A \vdash \Delta'}{(v, e) : (\Gamma, \Gamma' \vdash \Delta, \Delta')}\, cut$$

$$\frac{\Gamma, x : A \vdash v : B \mid \Delta}{\Gamma \vdash \lambda x.v : A \to B \mid \Delta} \to_R \qquad \frac{\Gamma \vdash v : A \mid \Delta \qquad \Gamma'; E : B \vdash \Delta'}{\Gamma, \Gamma'; v.E : A \to B \vdash \Delta, \Delta'} \to_L$$

$$\frac{\Gamma; E : A \vdash \Delta \qquad \Gamma' \vdash v : B \mid \Delta'}{\Gamma, \Gamma' \vdash E.v : B - A \mid \Delta, \Delta'}\,-_R \qquad \frac{\Gamma \mid e : B \vdash \beta : A, \Delta}{\Gamma; \beta\lambda.e : B - A \vdash \Delta}\,-_L$$

$$\frac{\Gamma; E : A \vdash \Delta}{\Gamma \mid E : A \vdash \Delta}\,de \qquad \frac{c : (\Gamma, x : A \vdash \Delta)}{\Gamma \mid \tilde{\mu}x.c : A \vdash \Delta}\,\tilde{\mu} \qquad \frac{c : (\Gamma \vdash \beta : B, \Delta)}{\Gamma \vdash \mu\beta.c : B \mid \Delta}\,\mu$$

$$\frac{\Gamma \mid \Pi \vdash \Pi \mid \Delta}{\Gamma \mid \Pi \vdash \Pi \mid \Delta, A}\,wk_R \qquad \frac{\Gamma \mid \Pi \vdash \Pi \mid \Delta}{\Gamma, A \mid \Pi \vdash \Pi \mid \Delta}\,wk_L$$

$$\frac{\Gamma \mid \Pi \vdash \Pi \mid \Delta, A, A}{\Gamma \mid \Pi \vdash \Pi \mid \Delta, A}\,co_R \qquad \frac{\Gamma, A, A \mid \Pi \vdash \Pi \mid \Delta}{\Gamma, A \mid \Pi \vdash \Pi \mid \Delta}\,co_L$$

Ce qui signifie que l'on a les règles structurelles sur les contextes pour tous les types de jugements.

On traduit le $\bar{\lambda}\mu\tilde{\mu}_T$-calcul dans $\mathsf{LLP}$ par la traduction $(.)^-$ en traduisant les jugements par :

$$\begin{array}{rcl}
c : (\Gamma \vdash \Delta) & \rightsquigarrow & \vdash ?\Gamma^{-\perp}, \Delta^- \\
\Gamma \vdash v : A \mid \Delta & \rightsquigarrow & \vdash ?\Gamma^{-\perp}, \Delta^-, A^- \\
\Gamma; E : A \vdash \Delta & \rightsquigarrow & \vdash A^{-\perp}, ?\Gamma^{-\perp}, \Delta^- \\
\Gamma \mid e : A \vdash \Delta & \rightsquigarrow & \vdash ?A^{-\perp}, ?\Gamma^{-\perp}, \Delta^-
\end{array}$$

En composant cette traduction avec la traduction $(.)^b$, on retrouve la traduction du $\bar{\lambda}\mu\tilde{\mu}_T$-calcul dans $\mathsf{LL}$ décrite par Curien-Herbelin.

12.4. Proposition (Simulation)
*Par la traduction $(.)^-$, la réduction de* $\mathsf{LLP}$ *simule celle du $\bar{\lambda}\mu\tilde{\mu}_T$-calcul.*

Démonstration : Dans la mesure où la réduction du $\bar{\lambda}\mu\tilde{\mu}_T$-calcul correspond à l'élimination des coupures de $\mathsf{LKT}$, il s'agit essentiellement d'un corollaire de la proposition 12.2. □

12.5. Remarque : Les $\bar{\lambda}\mu\tilde{\mu}_Q$-calcul et $\bar{\lambda}\mu\tilde{\mu}_T$-calcul cherchent à faire apparaître au maximum la symétrie entre appel par valeur et appel par nom, mais on voit que le point clef qui ne peut pas être rendu symétrique est la règle de déréliction.

Ceci termine notre description de traductions de systèmes classiques dans $\mathsf{LLP}$. Il n'y a, à notre connaissance, actuellement aucun calcul classique déterministe qui ne puisse pas être ainsi traduit. Ceci conforte l'idée que $\mathsf{LLP}$ puisse servir de cadre unificateur à l'étude de ces systèmes.

# Quatrième partie

# Analyse linéaire de la logique classique

La partie précédente a permis de montrer l'expressivité de LLP, et comment les systèmes classiques déterministes connus se traduisent dans LLP. On s'intéresse désormais aux apports de cette analyse dans LLP pour les différents systèmes. Nous développons trois points où ces apports sont particulièrement marquants.

Nous analysons tout d'abord les $\neg\neg$-traductions (ou CPS-traductions) à travers la logique linéaire, ce qui confirme le statut de système classique de LLP par rapport à LL ayant un statut plutôt intuitionniste.

Nous introduisons ensuite la notion de *logique classique linéaire* du point de vue syntaxique et sémantique, dont l'objectif est de montrer que la structure importante des preuves de logique classique peut en grande partie être étudiée en l'absence de règles structurelles (contraction et affaiblissement).

Enfin nous explicitons les rapports entre dualité appel par nom/appel par valeur et LLP, donnant une interprétation logique de ces deux notions fondamentales dans les langages de programmation.

# Chapitre 13

# CPS-traductions

La possibilité de traduire aussi bien la logique intuitionniste que la logique classique dans des systèmes linéaires offre un cadre pour l'analyse des CPS-traductions. Il apparaît que celles-ci correspondent à des traductions de LLP dans LL que l'on a naturellement rencontrées lors de l'étude de LLP. De plus le cas linéaire suggère des optimisations possibles pour les CPS-traductions.

Ce chapitre est le produit d'un travail en collaboration avec L. Regnier [LR02].

De manière à prendre en compte certaines optimisations de linéarité, on considère un $\lambda$-calcul enrichi avec un type $\neg_o A$ correspondant aux constructions suivantes :

$$\frac{\Gamma, x : A \vdash u : \bot}{\Gamma \vdash \ell x.u : \neg_o A}\ x \text{ linéaire} \qquad\qquad \frac{\Gamma \vdash u : \neg_o A \qquad \Delta \vdash v : A}{\Gamma, \Delta \vdash (u)^\ell v : \bot}$$

La signification de «$x$ linéaire», qui permet l'introduction du connecteur $\neg_o$, reste à préciser selon le contexte (par nom ou par valeur).

## 13.1 Traduction de Plotkin par nom

La *CPS-traduction par nom de Plotkin* [Plo75] (légèrement corrigée dans [HD97]) s'étend naturellement en une traduction du $\lambda\mu$-calcul dans le $\lambda$-calcul, que nous rappelons ici en introduisant des informations de linéarité.

La contrainte «$x$ linéaire» signifie, pour l'appel par nom, que $x$ apparaît exactement une fois dans le terme et pas en position d'argument.

Traduction des types $(.)^n$ : (où $\neg A$ qui équivaut à $A \to \bot$ est à considérer comme une version non linéaire de $\neg_o A$)

$$\begin{array}{ccc} X & \rightsquigarrow & X \\ A \to B & \rightsquigarrow & \neg\neg_o A^n \to \neg\neg_o B^n \end{array}$$

$$\Gamma \vdash A \mid \Delta \ \rightsquigarrow\ \neg\neg_o \Gamma^n, \neg_o \Delta^n \vdash \neg\neg_o A^n$$

Traduction des termes :

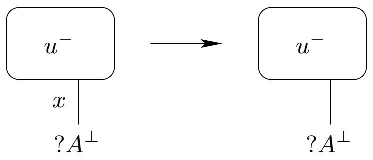
$$\begin{array}{lcl} x & \rightsquigarrow & \lambda k^{\neg_o A^n}.(x^{\neg\neg_o A^n})k \\ \lambda x.u & \rightsquigarrow & \lambda k^{\neg_o (A\to B)^n}.(k)^\ell \lambda x^{\neg\neg_o A^n}.u^{\neg\neg_o B^n} \\ (u)v & \rightsquigarrow & \lambda k^{\neg_o B^n}.(u^{\neg\neg_o (A\to B)^n})\ell m^{(A\to B)^n}.(m)v^{\neg\neg_o A^n} k \\ \mu\alpha.u & \rightsquigarrow & \lambda \alpha^{\neg_o A^n}.u^\bot \\ [\alpha]u & \rightsquigarrow & (u^{\neg\neg_o A^n})\alpha^{\neg_o A^n} \end{array}$$

on a omis les $(.)^n$ dans les termes traduits pour alléger l'écriture.

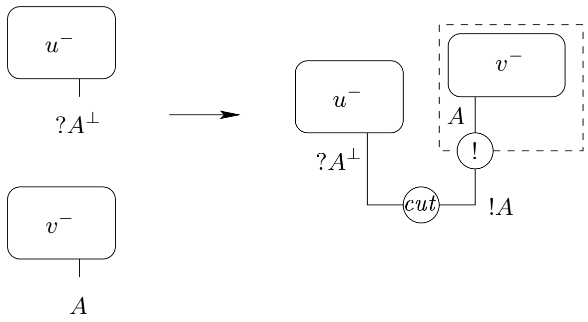
13.1. Remarque : Le remplacement de certains $\neg$ par des $\neg_o$ vient du fait que, dans la traduction d'un terme, une continuation $k$ (ou $\alpha$) utilise toujours son argument de manière linéaire.

Si on raffine la traduction $(.)^-$ du $\lambda$-calcul dans LL avec $(\neg_o A)^- = A^{-\perp}$ et en omettant les conclusions $\perp$ :

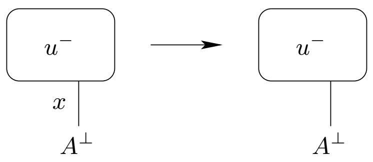
$\lambda x^A.u^\perp$



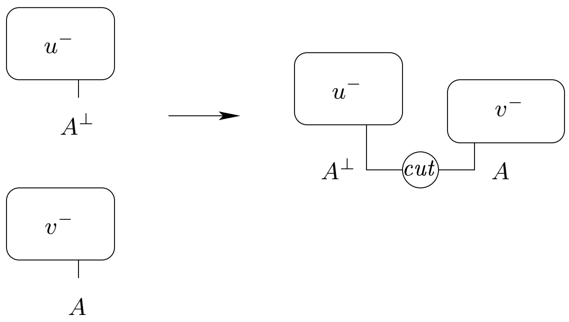

$(u^{\neg A})v^A$




$\ell x^A.u^\perp$




$(u^{\neg_o A})^\ell v^A$




on obtient le diagramme :

$$\begin{tikzcd}
\lambda\mu\text{-calcul} \arrow[r, "(.)^n"] \arrow[d, "(.)^-"'] & \lambda\text{-calcul} \arrow[d, "(.)^-"] \\
\mathsf{LLP} \arrow[r, "(.)^b"'] & \mathsf{LL}^+_{\text{pol}}
\end{tikzcd}$$

modulo réduction $ax$ et $\eta$-expansion.

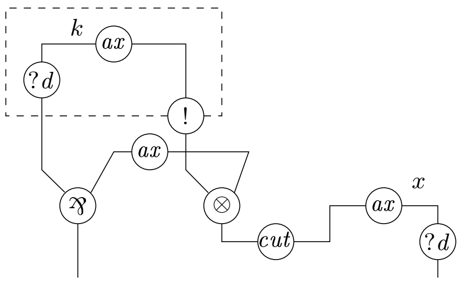
Démonstration : Regardons tout d'abord la traduction des jugements. Si on part du jugement $\Gamma \vdash A \mid \Delta$ du $\lambda\mu$-calcul et qu'on le traduit par $(.)^n$ on obtient $\neg\neg_o\Gamma^n, \neg_o\Delta^n \vdash \neg\neg_o A^n$ qui donne, par la traduction $(.)^-$, $\vdash ?!\Gamma^{n-\perp}, ?\Delta^{n-}, ?A^{n-}$. Si maintenant on traduit $\Gamma \vdash A \mid \Delta$ d'abord par $(.)^-$ ce qui donne $\vdash ?\Gamma^{-\perp}, A^-, \Delta^-$ puis par $(.)^b$, on obtient $\vdash ?\Gamma^{-b\perp}, A^{-b}, \Delta^{-b}$. Il nous reste à comparer $A^{n-}$ et $A^{-b}$ lorsque $A$ est un type du $\lambda\mu$-calcul. On va montrer que $A^{-b} = ?A^{n-}$ par récurrence sur la taille de $A$ :

$$\begin{aligned} X^{n-} &= X^- \\ &= X^\perp \\ X^{-b} &= X^{\perp b} \\ &= ?X^\perp \\ &= ?X^{n-} \end{aligned}$$

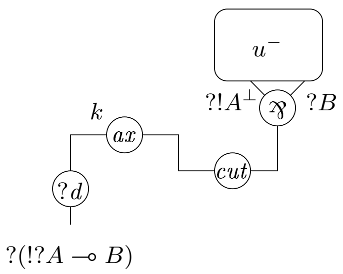
$$\begin{aligned} (A \to B)^{n-} &= (\neg\neg_o A^n \to \neg\neg_o B^n)^- \\ &= !?A^{n-} \multimap ?B^{n-} \\ (A \to B)^{-b} &= (!A^- \multimap B^-)^b \\ &= ?(!A^{-b} \multimap B^{-b}) \\ &= ?(!?A^{n-} \multimap ?B^{n-}) \\ &= ?(A \to B)^{n-} \end{aligned}$$

On a montré que le diagramme commute pour les jugements, on va maintenant le prouver pour les termes.

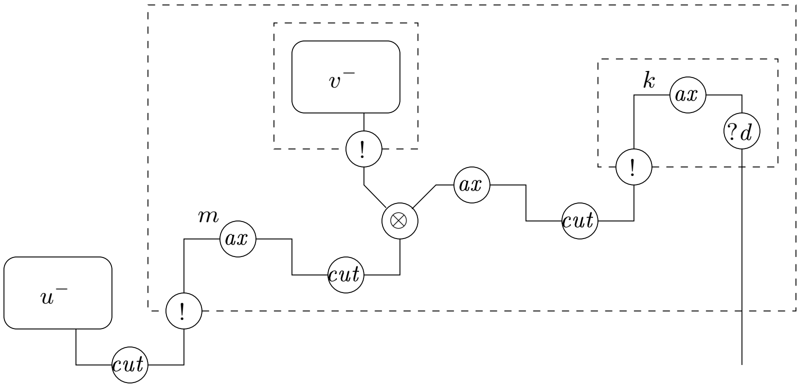
- La traduction du terme $\lambda k^{\neg_o A^n}.(x^{\neg\neg_o A^n})k$ donne le réseau :




  qui est, modulo réduction $ax$ et $\eta$-expansion, un axiome avec une dereliction, c'est-à-dire $(x^-)^b$.
- Par la traduction $(.)^-$, le terme $\lambda k^{\neg_o(\neg\neg_o A \to \neg\neg_o B)}.(k)^\ell \lambda x^{\neg\neg_o A}.u^{\neg\neg_o B}$ donne le réseau :




  qui est, après réduction $ax$, la traduction $(.)^b$ du réseau associé au $\lambda\mu$-terme $\lambda x.u$.
- Par la traduction $(.)^-$, le terme $\lambda k^{\neg_o B}.(u^{\neg\neg_o(\neg\neg_o A \to \neg\neg_o B)})\ell m^{\neg\neg_o A \to \neg\neg_o B}.(m)v^{\neg\neg_o A}k$ donne le réseau :




c'est-à-dire, modulo réduction $ax$ et $\eta$-expansion, la traduction $(.)^b$ du réseau associé au $\lambda\mu$-terme $(u)v$.

- Par la traduction $(.)^-$, le terme $\lambda\alpha^{\neg_o A}.u^\perp$ donne le même réseau que la traduction $(.)^-$ du terme $u$ si $\alpha$ est libre dans $u$ et ajoute un nœud $?w$ sinon, comme c'est le cas pour la traduction de $\mu\alpha.u$.
- Par la traduction $(.)^-$, le terme $(u^{\neg\neg_o A})\alpha^{\neg_o A}$ donne, modulo $\eta$-expansion et réduction $ax$, le même réseau que la traduction $(.)^-$ du terme $u$ si $\alpha$ n'est pas libre dans $u$ et ajoute un nœud $?c$ sinon, comme c'est le cas pour la traduction de $[\alpha]u$. $\square$

## 13.2 Traduction de Plotkin par valeur

On effectue un travail similaire sur la *traduction de Plotkin par valeur* (plus précisément sur la variante introduite par P. Selinger pour un langage avec paires [Sel01]). Cependant, dans le cas de l'appel par nom, l'optimisation de la traduction par des informations de linéarité est assez facile à appréhender alors qu'ici elle est plus difficile à percevoir. Il faut tout d'abord considérer le remplacement de certains $\neg$ par $\neg_o$ ainsi que les $\ell$ comme des annotations (le sens de «$x$ linéaire» tel qu'il apparaît dans la règle de typage de $\ell x.u$ n'étant pas le même que pour l'appel par nom).

Traduction des types $(.)^v$ :

$$\begin{array}{ccc}
X & \rightsquigarrow & X \\
A \to B & \rightsquigarrow & \neg(A^v \wedge \neg B^v) \\
A \wedge B & \rightsquigarrow & A^v \wedge B^v
\end{array}$$

$$\Gamma \vdash A \mid \Delta \quad \rightsquigarrow \quad \Gamma^v, \neg\Delta^v \vdash \neg_o\neg A^v$$

Traduction des termes :

$$\begin{array}{lcl}
x & \rightsquigarrow & \ell k^{\neg A^v}.(k)x^{A^v} \\
\lambda x.u & \rightsquigarrow & \ell k^{\neg(A\to B)^v}.(k)\lambda p^{A^v\wedge\neg B^v}.(u[{}^{\pi_1 p}/_x]^{\neg_o\neg B^v})^\ell \pi_2 p \\
(u)v & \rightsquigarrow & \ell k^{\neg B^v}.(u^{\neg_o\neg(A\to B)^v})^\ell \lambda m^{(A\to B)^v}.(v^{\neg_o\neg A^v})^\ell \lambda n^{A^v}.(m)<n,k> \\
\mu\alpha.u & \rightsquigarrow & \ell\alpha^{\neg A^v}.u^\perp \\
[\alpha]u & \rightsquigarrow & (u^{\neg_o\neg A^v})^\ell \alpha^{\neg A^v} \\
<u,v> & \rightsquigarrow & \ell k^{\neg(A\wedge B)^v}.(u^{\neg_o\neg A^v})^\ell \lambda m^{A^v}.(v^{\neg_o\neg B^v})^\ell \lambda n^{B^v}.(k)<m,n> \\
\pi_i u & \rightsquigarrow & \ell k^{\neg A_i^v}.(u^{\neg_o\neg(A_1\wedge A_2)^v})^\ell \lambda p^{A_1^v\wedge A_2^v}.(k)\pi_i p
\end{array}$$

On constate qu'à travers la traduction, les termes de type $\neg_o\neg A$ sont toujours utilisés comme fonction dans une application ($u$ dans la traduction de $\lambda x.u$ ou $u$ et $v$ dans la traduction de $(u)v$, ...).

C'est ce qui justifie l'optimisation de $\neg\neg A$ en $\neg_o\neg A$ pour ces termes. Cette optimisation fonctionne également si le terme est utilisé dans une $\lambda$-abstraction, bien que ce cas n'apparaisse pas ici.

Autrement dit, on peut également définir la traduction en mettant $\neg\neg A$ puis définir l'optimisation en remplaçant de manière uniforme tous les $\neg\neg A$ par $\neg_o\neg A$ si aucun n'est argument d'une application ni une variable. Ceci fonctionne à cause de la forme particulière des termes obtenus dans l'image de la traduction, l'optimisation n'est pas possible en général. Par exemple, le terme :

$$(\lambda y^{B\to\neg\neg A}.(t^{\neg\neg A\to C})(y)x^B)\lambda x^B.u^{\neg\neg A}$$

ne peut pas être optimisé puisque $(y)x$ de type $\neg\neg A$ apparaît comme argument d'une application, l'optimisation de $u^{\neg\neg A}$ en $u^{\neg_o\neg A}$ casserait le typage (en effet ce $(y)x$ va se réduire en $u$).

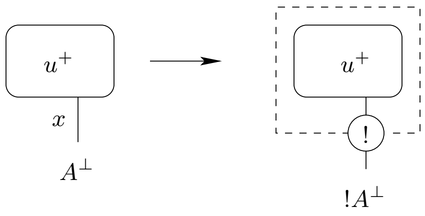
On considère comme traduction positive du $\lambda$-calcul dans $\mathsf{LL}$, la variante $(.)^\dagger$ de $(.)^+$ qui supprime les exponentiels liés à la logique classique (traduction de la flèche) et rajoute ceux nécessaires pour arriver dans $\mathsf{LL}$ (traduction de la conjonction). Cette traduction est décrite par exemple dans [DR99].

$$\begin{array}{ccc}
X & \rightsquigarrow & !X \\
A\to B & \rightsquigarrow & !(A^\dagger \multimap B^\dagger) \\
A\wedge B & \rightsquigarrow & !(A^\dagger \otimes B^\dagger) \\
\neg A & \rightsquigarrow & !A^{\dagger\perp} \\
\neg_o A & \rightsquigarrow & A^{\dagger\perp} \\
 & & \\
\Gamma \vdash A & \rightsquigarrow & \vdash \Gamma^{\dagger\perp}, A^\dagger
\end{array}$$

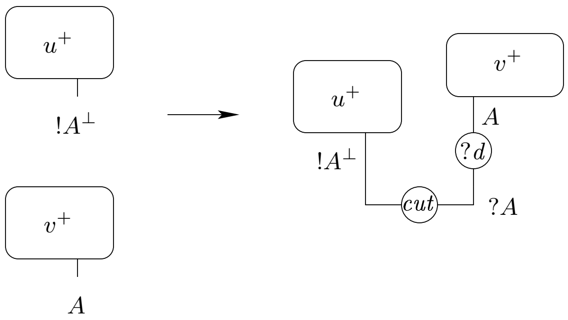
Cette traduction fonctionne parce que la traduction d'un type (excepté pour $\neg_o A$) commence toujours par un ! ce qui garantit la possibilité d'effectuer les règles structurelles intuitionnistes.

Pour l'abstraction et l'application des termes avec type $\neg A$, on utilise comme traduction :

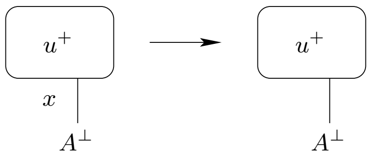
$\lambda x^A.u^\perp$



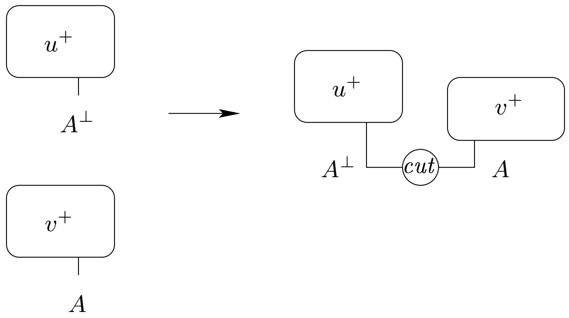

$(u^{\neg A})v^A$




$\ell x^A.u^\perp$

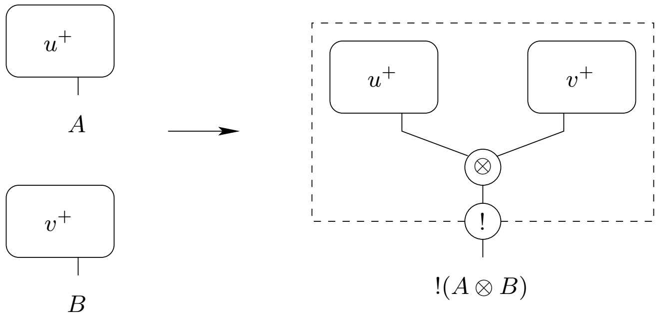


$(u^{\neg_o A})^{\ell} v^A$

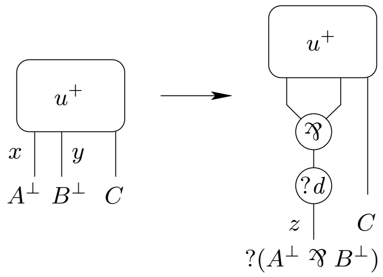



et pour les paires :

$<u^A, v^B>$




$u[{}^{\pi_1 z^{A\wedge B}}/_{x^A}, {}^{\pi_2 z^{A\wedge B}}/_{y^B}]^C$




On obtient le diagramme :

$$\begin{array}{ccc} \lambda\mu\text{-calcul} & \xrightarrow{(.)^v} & \lambda\text{-calcul} \\ {\scriptstyle (.)^+}\downarrow & & \downarrow{\scriptstyle (.)^\dagger} \\ \mathsf{LLP} & \xrightarrow[(.)^b]{} & \mathsf{LL}^+_{\mathrm{pol}} \end{array}$$

modulo réduction *ax*.

Démonstration : La traduction des jugements de chaque côté du diagramme donne :

$$\begin{array}{ccccc} \Gamma \vdash A \mid \Delta & \xrightarrow{(.)^v} & \Gamma^v, \neg\Delta^v \vdash \neg_o\neg A^v & \xrightarrow{(.)^\dagger} & \vdash \Gamma^{v\dagger\perp}, ?\Delta^{v\dagger}, ?A^{v\dagger} \\ \Gamma \vdash A \mid \Delta & \xrightarrow{(.)^+} & \vdash \Gamma^{+\perp}, ?A^+, ?\Delta^+ & \xrightarrow{(.)^b} & \vdash \Gamma^{+b\perp}, ?A^{+b}, ?\Delta^{+b} \end{array}$$

Pour que les résultats coïncident, il reste à montrer que $A^{v\dagger} = A^{+b}$. Ceci s'obtient par récurrence sur la taille de $A$ :

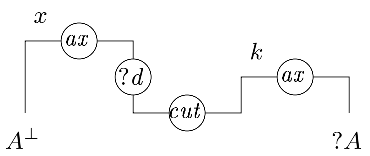
$$\begin{aligned}
X^{v\dagger} &= X^{\dagger}\\
&= !X\\
X^{+b} &= X^{b}\\
&= !X
\end{aligned}$$

$$\begin{aligned}
(A \to B)^{v\dagger} &= (\neg(A^v \wedge \neg B^v))^{\dagger}\\
&= !?(A^{v\dagger\perp} \parr ?B^{v\dagger})\\
(A \to B)^{+b} &= (!(A^{+} \multimap ?B^{+}))^{b}\\
&= !?(A^{+b} \multimap ?B^{+b})
\end{aligned}$$

$$\begin{aligned}
(A \wedge B)^{v\dagger} &= (A^v \wedge B^v)^{\dagger}\\
&= !(A^{v\dagger} \otimes B^{v\dagger})\\
(A \wedge B)^{+b} &= (A^{+} \otimes B^{+})^{b}\\
&= !(A^{+b} \otimes B^{+b})
\end{aligned}$$

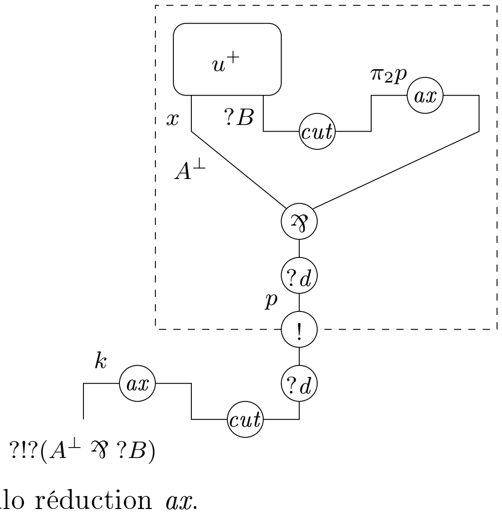
On va montrer maintenant que le diagramme commute pour les termes :

– Le $\lambda$-terme $\ell k^{\neg A}.(k)x^A$ donne, par la tradution $(.)^{\dagger}$, le réseau :



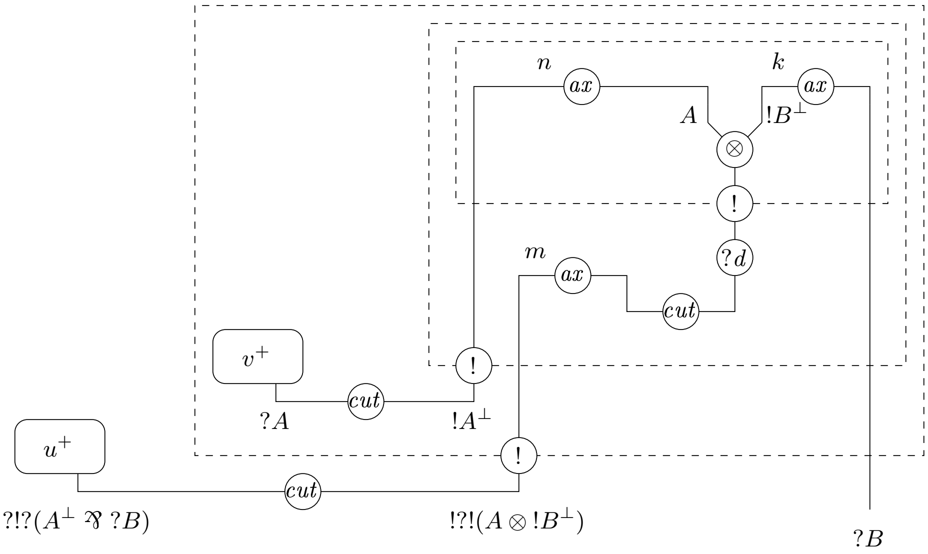

qui est bien celui que l'on obtient par $(x^{+})^{b}$.

– La traduction $(.)^{\dagger}$ du terme $\ell k^{\neg(A\wedge\neg B)}.(k)\lambda p^{A\wedge\neg B}.(u[{}^{\pi_1 p}/_{x^A}]^{\neg_o \neg B})^{\ell}\pi_2 p$ est le réseau :



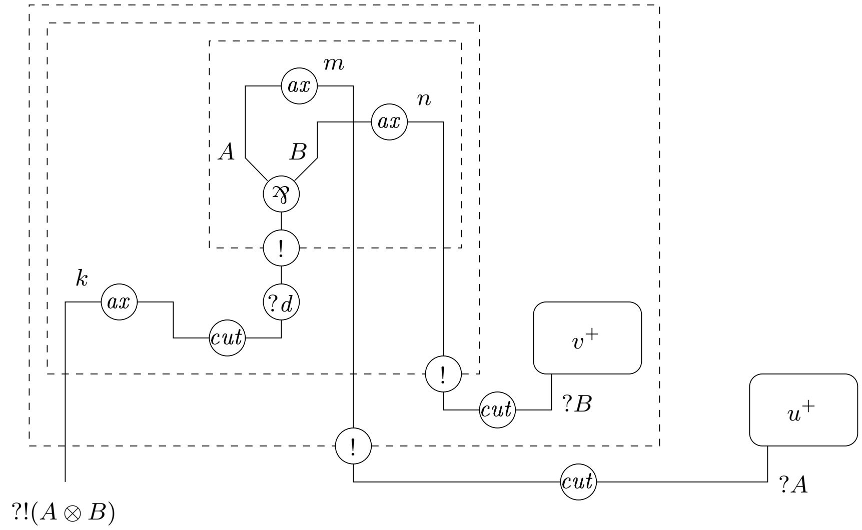

qui est $(\lambda x.u)^{+b}$ modulo réduction $ax$.

– La traduction $(.)^{\dagger}$ du terme $\ell k^{\neg B}.(u^{\neg_o \neg\neg(A\wedge\neg B)})^{\ell}\lambda m^{\neg(A\wedge\neg B)}.(v^{\neg_o \neg A})^{\ell}\lambda n^{A}.(m)<n,k>$ est le réseau :



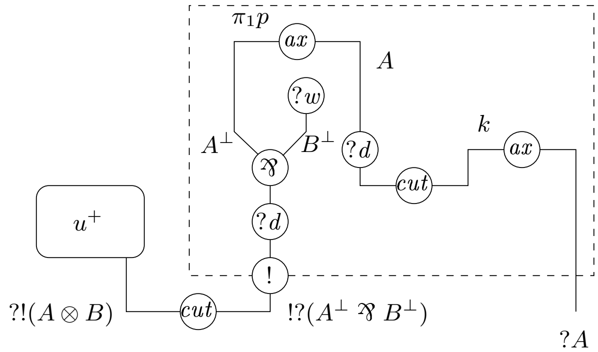

c'est-à-dire $((u)v)^{+b}$ (modulo réduction $ax$).

– La traduction $(.)^{\dagger}$ du terme $\ell k^{\neg(A\wedge B)}.(u^{\neg\circ\neg A})^{\ell}\lambda m^{A}.(v^{\neg\circ\neg B})^{\ell}\lambda n^{B}.(k){<}m,n{>}$ est le réseau :




c'est-à-dire $({<}u,v{>})^{+b}$ (modulo réduction $ax$).

– La traduction $(.)^{\dagger}$ du terme $\ell k^{\neg A}.(u^{\neg\circ\neg(A\wedge B)})^{\ell}\lambda p^{A\wedge B}.(k)\pi_1 p$ est le réseau :




c'est-à-dire $(\pi_1 u)^{+b}$ (modulo réduction $ax$).

- Le réseau associé au terme $\ell\alpha.u$ par $(.)^\dagger$ est le même que celui associé à $u$ si $\alpha$ est libre dans $u$ et nécessite l'ajout d'un affaiblissement sinon, comme c'est le cas pour $(\mu\alpha.u)^{+b}$.
- Le réseau associé au terme $(u)^\ell\alpha$ par $(.)^\dagger$ est le même que celui associé à $u$ si $\alpha$ n'est pas libre dans $u$ et nécessite l'ajout d'une contraction sinon, comme c'est le cas pour $([\alpha]u)^{+b}$.

□

## 13.3 Traduction de Krivine

On reprend la traduction du *fragment* $\forall, \to$ de la logique classique dans la logique intuitionniste donnée par J.-L. Krivine [Kri90] (des traductions similaires mais utilisant une double négation sont étudiées dans [Kle52]). On introduit cependant un raffinement en constatant des propriétés de linéarité (au sens de l'appel par nom).

Nous redonnons la preuve, due à J.-L. Krivine, des propriétés de cette traduction (lemmes 13.2 et 13.3 et proposition 13.4) car elle est utile pour faire le lien avec LLP.

La traduction des types est donnée par :

$$\begin{array}{ccc} X & \rightsquigarrow & \neg X \\ A \to B & \rightsquigarrow & A^r \to B^r \\ \forall X A & \rightsquigarrow & \forall X A^r \end{array}$$

Le séquent $\Gamma \vdash \Delta$ est traduit par $\Gamma^r, \neg_o \Delta^r \vdash$.

13.2. Lemme (Élimination des $\neg\neg_o$)
*Le séquent $\neg\neg_o A^r \vdash A^r$ est prouvable dans* LJ.

Démonstration : Par récurrence sur la taille de la formule $A$. On donne par ailleurs le $\lambda$-terme $c_A$ de type $\neg\neg_o A^r \to A^r$ correspondant.

$(X)$ $c_X = \lambda k'.\lambda x.(k')\ell k.(k)x$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{X \vdash X}}{X, \neg X \vdash}}{X \vdash \neg_o \neg X}}{X, \neg\neg_o \neg X \vdash}}{\neg\neg_o \neg X \vdash \neg X}$$

$(\to)$ $c_{A\to B} = \lambda k.\lambda a.(c_B)\lambda b.(k)\ell f.(b)^\ell(f)a$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A^r \vdash A^r} \quad \dfrac{}{B^r \vdash B^r}}{A^r, A^r \to B^r \vdash B^r}}{A^r, \neg_o B^r, A^r \to B^r \vdash}}{A^r, \neg_o B^r \vdash \neg_o(A^r \to B^r)}}{\neg\neg_o(A^r \to B^r), A^r, \neg_o B^r \vdash}}{\neg\neg_o(A^r \to B^r), A^r \vdash \neg\neg_o B^r} \qquad \neg\neg_o B^r \vdash B^r}{\neg\neg_o(A^r \to B^r), A^r \vdash B^r}}{\neg\neg_o(A^r \to B^r) \vdash A^r \to B^r}$$

(∀) $c_{\forall XA} = \lambda k.(c_A)\lambda f.(k)\ell a.(f)^{\ell} a$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A^r \vdash A^r}}{\forall X A^r \vdash A^r}}{\forall X A^r, \neg_o A^r \vdash}}{\neg_o A^r \vdash \neg_o \forall X A^r}}{\neg_o A^r, \neg\neg_o \forall X A^r \vdash}}{\neg\neg_o \forall X A^r \vdash \neg\neg_o A^r} \qquad \neg\neg_o A^r \vdash A^r}{\neg\neg_o \forall X A^r \vdash A^r}}{\neg\neg_o \forall X A^r \vdash \forall X A^r}$$

□

13.3. Lemme $((.)^r$ et substitution)
*Soient $A$ et $B$ deux formules, $A^r[{}^{\neg_o B^r}/_X] \vdash (A[{}^B/_X])^r$ est prouvable dans* LJ.

Démonstration : Par récurrence sur la taille de $A$ :
- Si $A = Y \neq X$, immédiat.
- Si $A = X$, $A^r = \neg X$ donc $A^r[{}^{\neg_o B^r}/_X] = \neg\neg_o B^r$ et $(A[{}^B/_X])^r = B^r$, or par le lemme 13.2, $\neg\neg_o B^r \vdash B^r$.
- Les autres cas sont de simples applications de l'hypothèse de récurrence. □

13.4. Proposition
*Si $\Gamma \vdash \Delta$ est prouvable dans* LK *alors $\Gamma^r, \neg_o \Delta^r \vdash$ est prouvable dans* LJ.

Démonstration : La traduction des règles structurelles ne pose aucun problème, il suffit de montrer comment on traduit les autres règles en appliquant les lemmes 13.2 et 13.3 :

$$\dfrac{\dfrac{}{X^r \vdash X^r}}{X^r, \neg_o X^r \vdash}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma^r, A^r, \neg_o \Delta^r, \neg_o B^r \vdash}{\Gamma^r, A^r, \neg_o \Delta^r \vdash \neg\neg_o B^r} \qquad \neg\neg_o B^r \vdash B^r}{\Gamma^r, A^r, \neg_o \Delta^r \vdash B^r}}{\Gamma^r, \neg_o \Delta^r \vdash A^r \to B^r}}{\Gamma^r, \neg_o \Delta^r, \neg_o (A^r \to B^r) \vdash}$$

$$\dfrac{\dfrac{\dfrac{\Gamma^r, \neg_o \Delta^r, \neg_o A^r \vdash}{\Gamma^r, \neg_o \Delta^r \vdash \neg\neg_o A^r} \qquad \neg\neg_o A^r \vdash A^r}{\Gamma^r, \neg_o \Delta^r \vdash A^r} \qquad \Gamma'^r, B^r, \neg_o \Delta'^r \vdash}{\Gamma^r, \Gamma'^r, \neg_o \Delta^r, \neg_o \Delta'^r, A^r \to B^r \vdash}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma^r, \neg_o \Delta^r, \neg_o A^r \vdash}{\Gamma^r, \neg_o \Delta^r \vdash \neg\neg_o A^r} \qquad \neg\neg_o A^r \vdash A^r}{\Gamma^r, \neg_o \Delta^r \vdash A^r}}{\Gamma^r, \neg_o \Delta^r \vdash \forall X A^r}}{\Gamma^r, \neg_o \Delta^r, \neg_o \forall X A^r \vdash}$$

$$\dfrac{\dfrac{\Gamma^r, (A[{}^B/_X])^r, \neg_o \Delta^r \vdash \qquad A^r[{}^{\neg_o B^r}/_X] \vdash (A[{}^B/_X])^r}{\Gamma^r, A^r[{}^{\neg_o B^r}/_X], \neg_o \Delta^r \vdash}}{\Gamma^r, \forall X A^r, \neg_o \Delta^r \vdash}$$

□

On peut traduire la logique intuitionniste dans $\mathsf{LL}$ par la *traduction négative* que l'on complète par :

$$\neg_o A \quad \rightsquigarrow \quad A^\perp$$

L'énoncé du lemme 13.2 devient, par traduction, $\vdash\, !A^\perp, A$ où $A$ est une formule négative. On constate que c'est le lemme 4.7 et que la preuve est basée sur la même récurrence et les mêmes preuves de calcul des séquents.

Reste à remarquer que si $A$ est une formule classique, $A^{-\rho} = A^{r-}$ et on obtient le diagramme suivant :

$$\begin{array}{ccc} \mathsf{LK} & \xrightarrow{(.)^r} & \mathsf{LJ} \\ {\scriptstyle (.)^-}\big\downarrow & & \big\downarrow{\scriptstyle (.)^-} \\ \mathsf{LLP} & \xrightarrow[(.)^\rho]{} & \mathsf{LL}^+_{\mathrm{pol}} \end{array}$$

qui montre que la traduction par *renversement* n'est pas quelque chose d'anodin mais plutôt de l'ordre d'une CPS-traduction puisqu'elle permet de simuler la traduction de Krivine de la logique classique dans la logique intuitionniste. Ceci conforte l'idée que $\mathsf{LLP}$ est un système classique alors que $\mathsf{LL}_{\mathrm{pol}}$ (ou $\mathsf{LL}^+_{\mathrm{pol}}$) est un système intuitionniste.

13.5. Remarque : Pour compléter le tableau, il conviendrait de définir, par dualité, une traduction de Krivine par valeur. On peut en effet obtenir une telle traduction en combinant renversement et traduction positive, cependant le langage cible est un système dual de $\mathsf{LJ}$ avec au plus une formule à gauche ce qui est peu standard.

# Chapitre 14

# Logique classique linéaire

La définition et l'étude des jeux polarisés suggèrent qu'il est intéressant de séparer les connecteurs exponentiels en deux parties $\uparrow\flat$ ou $\downarrow\sharp$ dont seule la deuxième est responsable des règles structurelles (cette idée provenant initialement de la ludique de Girard). On est ainsi amené à introduire un *fragment* de LLP n'utilisant pas les règles structurelles. On va voir que ce système permet de coder une partie linéaire de la logique classique qui est suffisante pour contenir la plupart des mécanismes de contrôle, rejetant l'idée que la caractéristique principale de la logique classique est liée aux règles structurelles.

## 14.1 MALLP

### 14.1.1 Formules polarisées linéaires

Les *formules polarisées linéaires* sont obtenues comme les *formules polarisées* mais en remplaçant les connecteurs exponentiels par les *décalages*.

On néglige ici les quantificateurs pour simplifier la présentation mais il n'y aurait aucune difficulté à les prendre en compte comme on l'a fait pour LLP.

$$\begin{array}{lclccccccccccc} P & ::= & X & \mid & P \otimes P & \mid & P \oplus P & \mid & 1 & \mid & 0 & \mid & \downarrow N \\ N & ::= & X^\perp & \mid & N \parr N & \mid & N \,\&\, N & \mid & \perp & \mid & \top & \mid & \uparrow P \end{array}$$

### 14.1.2 Règles

Les règles sont issues de celles de LLP en supprimant principalement les règles structurelles.

$$\frac{}{\vdash N, N^\perp}\,ax \qquad \frac{\vdash \Gamma, N \qquad \vdash N^\perp, \Delta}{\vdash \Gamma, \Delta}\,cut$$

$$\frac{\vdash \Gamma, N, M}{\vdash \Gamma, N \parr M}\,\parr \qquad \frac{\vdash \Gamma, P \qquad \vdash \Delta, Q}{\vdash \Gamma, \Delta, P \otimes Q}\,\otimes$$

$$\frac{\vdash \Gamma, N \qquad \vdash \Gamma, M}{\vdash \Gamma, N \,\&\, M}\,\& \qquad \frac{\vdash \Gamma, P}{\vdash \Gamma, P \oplus Q}\,\oplus_1 \qquad \frac{\vdash \Gamma, Q}{\vdash \Gamma, P \oplus Q}\,\oplus_2$$

$$\frac{\vdash \mathcal{N}, N}{\vdash \mathcal{N}, \downarrow N}\,\downarrow \qquad \frac{\vdash \Gamma, P}{\vdash \Gamma, \uparrow P}\,\uparrow$$

$$\dfrac{}{\vdash \Gamma, \top}\ \top \qquad\qquad \dfrac{\vdash \Gamma}{\vdash \Gamma, \bot}\ \bot \qquad\qquad \dfrac{}{\vdash 1}\ 1$$

$\mathcal{N}$ ne contient que des formules négatives et le contexte de la règle $\top$ contient au plus une formule positive.

14.1. Lemme (Réversibilité)
*Le connecteur $\downarrow$ est réversible mais le connecteur $\uparrow$ ne l'est pas.*

Démonstration : Il s'agit de la même preuve que pour la réversibilité de ! dans LLP :

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\pi\\ \vdash \Gamma, \downarrow N\end{array} \qquad \dfrac{\dfrac{}{\vdash N^\perp, N}\ ax}{\vdash \uparrow N^\perp, N}\ \uparrow}{\vdash \Gamma, N}\ cut}{}}{\vdash \Gamma, \downarrow N}\ \downarrow$$

Le séquent $\vdash \uparrow X, X^\perp$ n'étant pas prouvable puisqu'il contient deux formules positives alors que $\vdash \uparrow X, \downarrow X^\perp$ l'est, $\uparrow$ ne peut pas être réversible. □

En utilisant la *traduction négative* de la logique classique (section 9.2) où l'on remplace ! par $\downarrow$ et ? par $\uparrow$, on traduit dans MALLP un sous-système de la logique classique n'utilisant pas les règles structurelles.

Notation : De la même manière que $A \multimap B$ est une notation pour $A^\perp \parr B$, on utilisera $A \rightarrowtriangle B$ pour $\uparrow A^\perp \parr B$. Cette notation est particulièrement utile pour la traduction négative puisque $(A \to B)^- = A^- \rightarrowtriangle B^-$

14.2. Exemple
On peut traduire certaines tautologies classiques «linéaires» dans MALLP en utilisant la traduction négative :

$(\neg\neg A \to A)^- = \uparrow\downarrow A \rightarrowtriangle A$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A^\perp, A}\ ax}{\vdash \uparrow A^\perp, A}\ \uparrow}{\vdash \downarrow\uparrow A^\perp, A}\ \downarrow}{\vdash \uparrow\downarrow\uparrow A^\perp, A}\ \uparrow}{\vdash \uparrow\downarrow\uparrow A^\perp \parr A}\ \parr$$

$(A \to \neg\neg A)^- = A \rightarrowtriangle \uparrow\downarrow A$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A^\perp, A}\ ax}{\vdash \uparrow A^\perp, A}\ \uparrow}{\vdash \uparrow A^\perp, \downarrow A}\ \downarrow}{\vdash \uparrow A^\perp, \uparrow\downarrow A}\ \uparrow}{\vdash \uparrow A^\perp \parr \uparrow\downarrow A}\ \parr$$

$(\neg\neg A \to \neg\neg A)^- = \uparrow\downarrow A \rightarrowtriangle \uparrow\downarrow A$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A^\perp, A}\,ax}{\vdash \uparrow A^\perp, A}\,\uparrow}{\vdash \uparrow A^\perp, \downarrow A}\,\downarrow}{\vdash \uparrow A^\perp, \uparrow\downarrow A}\,\uparrow}{\vdash \downarrow\uparrow A^\perp, \uparrow\downarrow A}\,\downarrow}{\vdash \uparrow\downarrow\uparrow A^\perp, \uparrow\downarrow A}\,\uparrow}{\vdash \uparrow\downarrow\uparrow A^\perp \parr \uparrow\downarrow A}\,\parr \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A^\perp, A}\,ax}{\vdash \uparrow A^\perp, A}\,\uparrow}{\vdash \downarrow\uparrow A^\perp, A}\,\downarrow}{\vdash \uparrow\downarrow\uparrow A^\perp, A}\,\uparrow}{\vdash \uparrow\downarrow\uparrow A^\perp, \downarrow A}\,\downarrow}{\vdash \uparrow\downarrow\uparrow A^\perp, \uparrow\downarrow A}\,\uparrow}{\vdash \uparrow\downarrow\uparrow A^\perp \parr \uparrow\downarrow A}\,\parr$$

La première preuve est tout simplement une $\eta$-expansion d'un axiome, par contre la seconde qui est obtenue par élimination des coupures entre les preuves précédentes n'est pas du tout de la même nature ce qui correspond au fait que $\neg\neg A \not\simeq A$.

$(A \to A)^- = A \rightarrowtriangle A = \uparrow A^\perp \parr A = (\neg A \vee A)^-$

$$\dfrac{\dfrac{\dfrac{}{\vdash A^\perp, A}\,ax}{\vdash \uparrow A^\perp, A}\,\uparrow}{\vdash \uparrow A^\perp \parr A}\,\parr$$

14.3. Remarque : Les preuves de MALLP sont des décorations des preuves de MALL, en particulier, toute sémantique dénotationnelle de MALL est une sémantique dénotationnelle de MALLP avec $\downarrow A = A$ ce qui dégénère la structure de MALLP. En effet, la propriété cruciale de MALLP est de pouvoir exprimer une négation $\neg A = \uparrow A^\perp$ qui vérifie $\neg\neg A \not\simeq A$, tout en restant dans un cadre où règne une véritable dualité $A^{\perp\perp} = A$.

Concernant la syntaxe, toutes les solutions que nous avons présentées dans le chapitre 5 pour les réseaux polarisés peuvent s'appliquer à MALLP avec des boîtes pour le connecteur $\downarrow$.

## 14.2 $\lambda\mu$-calcul linéaire

On dit qu'un terme est linéaire si chaque variable a exactement une occurrence, en particulier pour les variables liées. Si on s'en tient aux définitions des annexes D.1 et D.2, on a le lemme suivant :

14.4. Lemme (Dégénérescence du $\lambda\mu$-calcul linéaire)
*Un $\lambda\mu$-terme linéaire est $\theta$-équivalent à un $\lambda$-terme linéaire.*

Démonstration : Par récurrence sur le nombre de $\mu$ dans $t$. S'il n'y en a pas le résultat est prouvé. Sinon on considère un sous-terme $\mu\alpha[\beta]u$ de $t$ tel qu'il n'y ait pas de $\mu$ dans $u$. $u$ étant un $\lambda$-terme, il ne contient ni $\alpha$ ni $\beta$ et pour que $\mu\alpha[\beta]u$ soit linéaire, il faut que $[\beta]u$ contienne exactement une occurrence de $\alpha$ donc $\alpha = \beta$, d'où $\mu\alpha[\beta]u \simeq_\theta u$. Par hypothèse de récurrence appliquée à $t[{}^{u}/_{\mu\alpha[\beta]u}]$, on en déduit que $t$ est $\theta$-équivalent à un $\lambda$-terme. $\square$

On considère donc, comme dans l'annexe D.5, des $\lambda\mu$-terme où les constructions $\mu\alpha$ et $[\alpha]$ sont indépendantes. Un tel $\lambda\mu$-terme linéaire n'est pas forcément un $\lambda$-terme comme le montrent les exemples suivants.

14.5. Exemple (Appel par nom)
Les $\lambda\mu$-termes linéaires correspondant aux preuves de l'exemple 14.2 sont :

$$\lambda f^{\neg\neg A}.\mu\alpha^{A}.(f)\lambda x^{A}.[\alpha]x : \neg\neg A \to A$$
$$\lambda x^{A}.\lambda k^{\neg A}.(k)x : A \to \neg\neg A$$
$$\lambda f^{\neg\neg A}.\lambda g^{\neg A}.(f)\lambda x^{A}.(g)x \simeq_{\eta} \lambda f^{\neg\neg A}.f : \neg\neg A \to \neg\neg A$$
$$\lambda f^{\neg\neg A}.\lambda g^{\neg A}.(g)\mu\alpha^{A}.(f)\lambda x^{A}.[\alpha]x : \neg\neg A \to \neg\neg A$$

14.6. Remarque : Le $\lambda\mu$-terme $\lambda f^{\neg\neg A}.\mu\beta^{\neg\neg A}.(f)\lambda x^{A}.[\beta]\lambda g^{\neg A}.(g)x : \neg\neg A \to \neg\neg A$ a la même traduction négative que le terme $\lambda f^{\neg\neg A}.\lambda g^{\neg A}(f)\lambda x^{A}.(g)x$ si on applique les simplifications décrites au chapitre précédent pour traduire $(u^{\neg A})v^{A}$ et $\lambda x^{A}.u^{\perp}$. Pourtant ils ne sont pas $\sigma$-équivalents pour les équations de la définition 11.18, en effet les simplifications correspondant au type $\neg A$ (ou $\perp$) introduisent de nouvelles équations dans la $\sigma$-équivalence.

14.7. Exemple (Appel par valeur)
Les $\lambda\mu$-termes linéaires en appel par valeur correspondant aux preuves de l'exemple 14.2 (modulo quelques règles finales uniquement déterminées par le type) sont :

– pour les deux premières preuves :

$$\lambda x^{A}.\lambda k^{\neg A}.(k)x : A \to \neg\neg A$$
$$\lambda f^{\neg\neg A}.\mu\alpha^{A}.(f)\lambda x^{A}.[\alpha]x : \neg\neg A \to A$$

– la première preuve de $\vdash \uparrow\downarrow\uparrow A^{\perp}, \uparrow\downarrow A$ est la traduction des deux termes suivants :

$$\lambda f^{\neg\neg A}.\lambda g^{\neg A}.(g)\mu\alpha^{A}.(f)\lambda x^{A}.[\alpha]x : \neg\neg A \to \neg\neg A$$
$$\lambda f^{\neg\neg A}.\lambda g^{\neg A}.(f)\lambda x^{A}.(g)x : \neg\neg A \to \neg\neg A$$

ce sont donc des $\lambda\mu$-termes «$\sigma$-équivalents pour la traduction positive».

– alors que la seconde preuve est la traduction du terme :

$$\lambda f^{\neg\neg A}.\mu\beta^{\neg\neg A}.(f)\lambda x^{A}.[\beta]\lambda g^{\neg A}.(g)x : \neg\neg A \to \neg\neg A$$

14.8. Remarque : On vient de voir un exemple des fortes différences entre la $\sigma$-équivalence par nom (correspondant à la traduction négative) et la $\sigma$-équivalence par valeur (correspondant à la traduction positive) puisque parmi les trois $\lambda\mu$-termes suivants :

$$\lambda f.\mu\beta.(f)\lambda x.[\beta]\lambda g.(g)x$$
$$\lambda f.\lambda g.(f)\lambda x.(g)x \simeq_{\eta} \lambda f.f$$
$$\lambda f.\lambda g.(g)\mu\alpha.(f)\lambda x.[\alpha]x$$

la traduction négative identifie les deux premiers et la traduction positive identifie les deux derniers.

Comme on en a vu des exemples, les traductions positives et négatives permettent d'envoyer le $\lambda\mu$-calcul linéaire dans MALLP en utilisant $\downarrow$ à la place de ! et $\uparrow$ à la place de ?. Ces traductions donnent lieu à des simulations de la réduction.

## 14.3 Ludique multiplicative et additive

Dans son article sur la ludique [Gir01], J.-Y. Girard décrit un système focalisé $\mathsf{MALL_f}$ pour MALL. Ce système ne possède aucune contrainte portant sur le nombre de formules positives contrairement à ce qui se passe dans MALLP. Nous allons voir que les contraintes de focalisation font qu'il est possible de traduire très directement $\mathsf{MALL_f}$ dans MALLP malgré cette différence.

### 14.3.1 Le système $\mathsf{MALL_f}$

Les formules de $\mathsf{MALL_f}$ sont les formules positives de $\mathsf{MALLP}$, les séquents sont de la forme $\Gamma \vdash \Delta; \Pi$ où $\Pi$ contient au plus une formule et si $\Pi$ est non vide alors $\Gamma$ ne contient que des atomes. Les règles sont les suivantes :

$$\frac{}{X \vdash ; X}\,ax \qquad \frac{\Gamma \vdash \Delta; P \qquad \Gamma', P \vdash \Delta'; \Pi}{\Gamma, \Gamma' \vdash \Delta, \Delta'; \Pi}\,cut \qquad \frac{\Gamma \vdash \Delta, P; \qquad \Gamma', P \vdash \Delta';}{\Gamma, \Gamma' \vdash \Delta, \Delta';}\,cut$$

$$\frac{\Gamma \vdash \Delta; P}{\Gamma \vdash \Delta, P;}\,de \qquad \frac{\Gamma \vdash \Delta, P;}{\Gamma, \downarrow P^{\perp} \vdash \Delta;}\uparrow \qquad \frac{\Gamma, P \vdash \Delta;}{\Gamma \vdash \Delta; \downarrow P^{\perp}}\downarrow$$

$$\frac{\Gamma \vdash \Delta; P \qquad \Gamma' \vdash \Delta'; Q}{\Gamma, \Gamma' \vdash \Delta, \Delta'; P \otimes Q}\otimes \qquad \frac{\Gamma, P, Q \vdash \Delta;}{\Gamma, P \otimes Q \vdash \Delta;}\parr$$

$$\frac{\Gamma \vdash \Delta; P}{\Gamma \vdash \Delta; P \oplus Q}\oplus_1 \qquad \frac{\Gamma \vdash \Delta; Q}{\Gamma \vdash \Delta; P \oplus Q}\oplus_2 \qquad \frac{\Gamma, P \vdash \Delta; \qquad \Gamma, Q \vdash \Delta;}{\Gamma, P \oplus Q \vdash \Delta;}\,\&$$

$$\frac{}{\vdash ; 1}\,1 \qquad \frac{\Gamma \vdash \Delta;}{\Gamma, 1 \vdash \Delta;}\perp \qquad \frac{}{\Gamma, 0 \vdash \Delta;}\top$$

14.9. Remarque : Ce système est quasiment identique au système obtenu en $\eta$-expansant les axiomes dans $\mathsf{LK}^{\eta}_{\mathrm{pol}}$ et en supprimant les règles structurelles.

### 14.3.2 $\mathsf{MALL_f}$ et $\mathsf{MALLP}$

Pour traduire $\mathsf{MALL_f}$ dans $\mathsf{MALLP}$, il suffit de traduire les formules par elles-mêmes et le séquent $\Gamma \vdash \Delta; \Pi$ par $\vdash \Gamma^{\perp}, \uparrow\Delta, \Pi$. Les règles sont alors immédiates à traduire.

Il est facile de voir que cette traduction donne une simulation de l'élimination des coupures de $\mathsf{MALL_f}$ par celle de $\mathsf{MALLP}$.

14.10. Proposition (Modèle de jeux pour $\mathsf{MALL_f}$)
*Les jeux polarisés fournissent un modèle dénotationnel de* $\mathsf{MALL_f}$.

On a le choix entre la traduction qui précède ou une définition directe de l'interprétation de $\mathsf{MALL_f}$ dans les jeux polarisés qui ne nécessite pas d'ajouter de $\uparrow$ systématiques sur les formules positives. En effet on peut interpréter le séquent $\Gamma \vdash \Delta; \Pi$ par une stratégie sur $(\odot\Delta^{\perp}) \odot \Pi^{\perp} \multimap (\parr\, \Gamma^{\perp})$ centrale en $\Pi^{\perp}$ s'il est non vide.

14.11. Remarque : On montrerait réciproquement que tout réseau de $\mathsf{MALLP}$ peut se séquentialiser en une preuve de $\mathsf{MALL_f}$.

La possibilité de traduire le $\lambda\mu$-calcul linéaire et $\mathsf{MALL_f}$ dans $\mathsf{MALLP}$ montre que $\mathsf{MALLP}$ n'est pas un système ad hoc mais correspond bien à une notion générale de logique classique linéaire. On pourrait traduire de même $\mathsf{LC}$ (ou $\mathsf{LK}^{\eta}_{\mathrm{pol}}$) sans règle structurelle.

## 14.4 Catégories de contrôle linéaires

Nous allons désormais chercher une notion catégorique de modèle de la logique classique linéaire. Le point de départ étant le modèle des jeux polarisés, les catégories que l'on va définir posséderont un constructeur $\odot$ qui n'est pas défini dans la syntaxe de MALLP, il ne s'agit donc pas de la notion la plus générale possible de modèle catégorique de MALLP.

La notion de catégorie faiblement distributive a été introduite par Cockett-Seely [CS97] pour étudier les modèles de MLL et de son fragment intuitionniste. On pourra également se référer à [BCST96].

14.12. Définition (Catégorie faiblement distributive symétrique)
Une *catégorie faiblement distributive symétrique* est une catégorie munie de deux bifoncteurs (produits tensoriels) $\odot$ d'élément neutre $I$ et $⅋$ d'élément neutre $\bot$, avec les isomorphismes naturels suivants :
- $a_\odot : (A \odot B) \odot C \xrightarrow{\sim} A \odot (B \odot C)$
- $s_\odot : A \odot B \xrightarrow{\sim} B \odot A$
- $u_\odot : A \odot I \xrightarrow{\sim} A$
- $a_⅋ : (A ⅋ B) ⅋ C \xrightarrow{\sim} A ⅋ (B ⅋ C)$
- $s_⅋ : A ⅋ B \xrightarrow{\sim} B ⅋ A$
- $u_⅋ : A ⅋ \bot \xrightarrow{\sim} A$

ainsi que la transformation naturelle de *faible distributivité* :
- $\delta_{A,B,C} : A \odot (B ⅋ C) \to (A \odot B) ⅋ C$

Ces transformations naturelles doivent vérifier un ensemble de conditions de cohérence que nous ne rappellerons pas ici.

Ajouter la négation revient à ajouter une fonction $\neg$ des objets dans les objets avec les morphismes suivants :
- $\gamma_A : A \odot \neg A \to \bot$
- $\tau_A : I \to \neg A ⅋ A$

vérifiant la condition de cohérence :

$$id_A \odot \tau_A ; \delta_{A,\neg A,A} ; \gamma_A ⅋ id_A ; u_⅋ = u_\odot$$

14.13. Proposition
*La notion de catégorie faiblement distributive symétrique avec négation coïncide avec celle de catégorie $\star$-autonome.*

14.14. Définition (Catégorie pré faiblement distributive symétrique)
Une *catégorie pré faiblement distributive symétrique* est une catégorie faiblement distributive symétrique dans laquelle le second produit tensoriel n'est pas un bifoncteur mais seulement un foncteur binoïdal [PR97] (on notera $\odot$ le produit tensoriel et $⅋$ le tenseur binoïdal). Les isomorphismes naturels $a_\odot$, $s_\odot$, $u_\odot$, $a_⅋$, $s_⅋$ et $u_⅋$ doivent être *centraux* ainsi que la transformation naturelle $\delta_{A,B,C}$.

Une catégorie pré faiblement distributive symétrique avec négation n'est pas nécessairement $\star$-autonome puisqu'on n'a pas en général $A \simeq \neg\neg A$, contrairement à ce qu'il se passe pour les catégories faiblement distributives du fait que $⅋$ est bifonctoriel.

14.15. Proposition
*Une catégorie pré faiblement distributive symétrique avec négation est une* catégorie monoïdale symétrique fermée *avec* $B^A = \neg A ⅋ B$.

Démonstration : Soit $f : A \odot B \to C$, on lui associe $f' : A \to C^B$ défini par $f' = u_\odot ; A \odot \tau_B ; \delta_{A,B,\neg B} ; f ⅋ \neg B$.

Réciproquement, on note $ev : C^B \odot B \to C$ le morphisme défini par $ev = \delta_{B,\neg B,C} ; \gamma_A ⅋ C ; u_⅋$ et si $g : A \to C^B$, on définit $g_0 : A \odot B \to C$ par $g_0 = g \odot B ; ev$.

Ces deux fonctions sont des bijections inverses l'une de l'autre. En effet, le diagramme suivant commute :

$$
\begin{tikzcd}
A\odot B \arrow[d, "A\odot u_\odot"'] & & \\
A\odot I\odot B \arrow[d, "A\odot\tau_B\odot B"'] & & \\
A\odot(B\parr\neg B)\odot B \arrow[r, "\delta_{A,B,\neg B}\odot B"] \arrow[d, "A\odot\delta_{B,\neg B,B}"'] & ((A\odot B)\parr\neg B)\odot B \arrow[r, "(f\parr\neg B)\odot B"] \arrow[d, "\delta_{B,\neg B,A\odot B}"'] & (C\parr\neg B)\odot B \arrow[d, "\delta_{B,\neg B,C}"] \\
A\odot(B\parr(\neg B\odot B)) \arrow[r, "\delta_{A,B,\neg B\odot B}"] \arrow[d, "A\odot(B\parr\gamma_B)"'] & (A\odot B)\parr(\neg B\odot B) \arrow[r, "f\parr(\neg B\odot B)"] \arrow[d, "(A\odot B)\parr\gamma_B"'] & C\parr(\neg B\odot B) \arrow[d, "C\parr\gamma_B"] \\
A\odot(B\parr\bot) \arrow[r, "\delta_{A,B,\bot}"'] & (A\odot B)\parr\bot \arrow[r, "f\parr\bot"'] \arrow[d, "u_\parr"'] & C\parr\bot \arrow[d, "u_\parr"] \\
& A\odot B \arrow[r, "f"'] & C
\end{tikzcd}
$$

par lignes :
- condition de cohérence de $\delta_{A,B,C}$
- naturalité de $\delta_{A,B,C}$
- naturalité de $\delta_{A,B,C}$
- centralité de $\gamma_A$
- naturalité de $u_\parr$

La partie supérieure droite est la définition de $(f')_0$ et la partie inférieure gauche est $f$ par la condition de cohérence de $\neg$.

Le diagramme suivant commute également :

$$
\begin{tikzcd}
A \arrow[r, "u_\odot"] \arrow[d, "g"'] & A\odot I \arrow[r, "A\odot\tau_B"] \arrow[d, "g\odot I"'] & A\odot(B\parr\neg B) \arrow[r, "\delta_{A,B,\neg B}"] \arrow[d, "g\odot(B\parr\neg B)"'] & (A\odot B)\parr\neg B \arrow[d, "(g\odot B)\parr\neg B"'] \\
C\parr\neg B \arrow[r, "u_\odot"] & (C\parr\neg B)\odot I \arrow[r, "(C\parr\neg B)\odot\tau_B"] \arrow[d, "\delta_{I,\neg B,C}"'] & (C\parr\neg B)\odot(B\parr\neg B) \arrow[r, "\delta_{(C\parr\neg B),B,\neg B}"] \arrow[d, "\delta_{B\parr\neg B,\neg B,C}"'] & ((C\parr\neg B)\odot B)\parr\neg B \arrow[d, "\delta_{B,\neg B,C}\parr\neg B"'] \\
& C\parr(\neg B\odot I) \arrow[r, "C\parr(\neg B\odot\tau_B)"'] & C\parr(\neg B\odot(B\parr\neg B)) \arrow[r, "C\parr\delta_{\neg B,B,\neg B}"'] & C\parr(\neg B\odot B)\parr\neg B \arrow[d, "C\parr\gamma_B\parr\neg B"'] \\
& & & C\parr\bot\parr B \arrow[d, "C\parr u_\parr"'] \\
& & & C\parr B
\end{tikzcd}
$$

par lignes :
- naturalité de $u_\odot$

– fonctorialité de $\odot$
– naturalité de $\delta_{A,B,C}$
– naturalité de $\delta_{A,B,C}$
– condition de cohérence de $\delta_{A,B,C}$
La partie supérieure droite est la définition de $(g_0)'$ et la partie inférieure gauche est $g$ par la condition de cohérence de $\neg$. □

L'introduction de l'exponentielle autorise à noter $\perp^A \simeq \neg A$ puisque $\perp^A = \neg A \parr \perp$ et $\neg A \parr \perp \simeq \neg A$ par $u_{\parr}$.

14.16. Définition (Catégorie de contrôle linéaire)
Une *catégorie de contrôle linéaire* est une catégorie pré faiblement distributive symétrique avec négation qui est également cartésienne et distributive.

Dans une catégorie de contrôle linéaire, on peut définir les morphismes suivants :
– $s_{A,B,C} = a_{\parr} : B^A \parr C \xrightarrow{\sim} (B \parr C)^A$ par $B^A = \neg A \parr B$,
– $p_{A,B,C} : B^A \overset{\bullet}{\to} (B \parr C)^{A \parr C}$ est le curryfié de $\delta_{B^A,A,C}; ev \parr id_C : B^A \odot (A \parr C) \to B \parr C$,
– $\theta_A = p_{\perp^A,\perp,A}; (\perp \parr A)^{\tau_A}; u_{\odot}; u_{\parr} : \perp^{\perp^A} \to A$.

14.17. Remarque : Si l'on est en mesure de définir les catégories de contrôle linéaires qui sont basées sur le morphisme $\delta_{A,B,C}$, c'est parce que dans l'axiomatique des catégories de contrôle, le morphisme $d'_{A,B,C}$ est réellement le point important, plus que la vraie distributivité $d_{A,B,C}$.

14.18. Proposition (Catégorie de contrôle linéaire de jeux)
*Les jeux AJM polarisés (sans relation d'équivalence) forment une catégorie de contrôle linéaire en prenant $I = \top$ et $\neg A = \uparrow A^{\perp}$.*

Démonstration : Il n'est pas utile d'introduire la relation d'équivalence entre parties d'après la remarque 7.44.
En utilisant ce que l'on a déjà vu dans les chapitres 3 et 7, il nous suffit de définir la stratégie $\delta_{A,B,C} : A \odot (B \parr C) \overset{\bullet}{\twoheadrightarrow} (A \odot B) \parr C$.
Soit $s$ est une partie du jeu $(A \odot B) \parr C$, on note $\tilde{s}$ la partie de $A \odot (B \parr C)$ obtenue de la manière suivante :
– si le premier coup de $s$ est de la forme $(a, c)$, on le remplace par $a$ et on remplace le premier coup dans $b$ (s'il existe) par $(b, c)$, les autre coups restent inchangés ;
– si le premier coup de $s$ est de la forme $(b, c)$, $\tilde{s} = s$.
On définit $\delta_{A,B,C} = \{s \in \mathcal{P}^J_{A\odot(B\parr C)\twoheadrightarrow(A\odot B)\parr C} \mid \forall t \leq^J s, t \upharpoonright_{A\odot(B\parr C)} = \widetilde{(t \upharpoonright_{(A\odot B)\parr C})}\}$.
On vérifie alors que les axiomes de catégorie de contrôle linéaire sont satisfaits. □

Ce résultat montre que l'on peut ne pas avoir $A \simeq \neg\neg A$ dans une catégorie de contrôle linéaire. Cependant une catégorie $\star$-autonome est une catégorie de contrôle linéaire, il y a donc des catégories de contrôle linéaires avec $A \simeq \neg\neg A$ qui ne sont pas dégénérées (*i.e.* telles qu'il existe deux objets avec plus d'un morphisme entre eux). Dans le cas des catégories de contrôle, si $A \simeq \neg\neg A$ (ou, de manière équivalente, si $\parr$ est un bifoncteur), on obtient une catégorie dégénérée (algèbre de Boole).

14.19. Remarque : Dans la catégorie de contrôle linéaire des jeux polarisés, on a $I = \top$. Ceci n'est a priori pas imposé par l'axiomatique des catégories de contrôle linéaires (même si on a $I \vdash \top$ puisque $\top$ est terminal). Cette propriété est-elle vérifiée dans beaucoup d'exemples concrets et devrait-elle être imposée dans l'axiomatique ? Nous reviendrons sur cette question dans le cadre de la logique classique affine.

14.20. Exemple (Non définissabilité)
Il n'y a pas de résultat de définissabilité direct ici à cause de l'existence de stratégies correspondant à des règles structurelles sur de petits types. La stratégie formée des deux parties suivantes (et de leurs préfixes), sur le jeu $\uparrow 1 \parr \uparrow 1 \multimap \uparrow 1$, est une pseudo-contraction qui n'a pas de contre-partie syntaxique :

$$\begin{array}{ccccccccc}
( & \uparrow 1 & \parr & \uparrow 1 & ) & \multimap & \uparrow 1 & & \\
& & & & & & \star & & O \\
( & \star & , & \star & ) & & & & J \\
& 1 & & & & & & & O \\
& & & & & & 1 & & J
\end{array}$$

et

$$\begin{array}{ccccccccc}
& & & & & & \star & & O \\
( & \star & , & \star & ) & & & & J \\
& & & 1 & & & & & O \\
& & & & & & 1 & & J
\end{array}$$

Introduire une contrainte explicite de linéarité comme dans [Gir01] permetrait certainement d'obtenir un résultat de définissabilité.

14.21. Lemme (Réversibilité)
*Soit $\mathbb{A}$ une catégorie de contrôle linéaire,* $\mathbb{A}(I, A \parr B) \simeq \mathbb{A}^\bullet(\neg A, B)$.

Démonstration : Soit $f : I \to A \parr B$, on note $f' : \neg A \xrightarrow{\bullet} B$ le morphisme central défini par $f' = p_{A,\perp,B}; (\perp \parr B)^f; u_\parr$.

Soit $g : \neg A \xrightarrow{\bullet} B$, on note $g_0 : I \to A \parr B$ le morphisme défini par $g_0 = \tau_A; A \parr g$.

Ces deux bijections sont inverses l'une de l'autre, même preuve que pour le lemme 6.19. □

14.22. Définition (Interprétation de MALLP)
Une formule négative $N$ est interprétée par un objet $[\![N]\!]_{cl}$ de la même manière que dans la définition 6.20 avec :

$$\uparrow N^\perp \quad \rightsquigarrow \quad \neg [\![N]\!]_{cl}$$

Une preuve $\pi$ est interprétée par un morphisme $[\![\pi]\!]_{cl}$ de la catégorie avec deux cas selon la forme du séquent conclusion :

$$\begin{array}{lcl}
\vdash N_1, \dots, N_n & \rightsquigarrow & [\![\pi]\!]_{cl} : \top \to [\![N_1]\!]_{cl} \parr \dots \parr [\![N_n]\!]_{cl} \\
\vdash N_1, \dots, N_n, P & \rightsquigarrow & [\![\pi]\!]_{cl} : [\![P^\perp]\!]_{cl} \xrightarrow{\bullet} [\![N_1]\!]_{cl} \parr \dots \parr [\![N_n]\!]_{cl}
\end{array}$$

en prenant $\perp$ à droite si $n = 0$.

Les règles sont interprétées comme dans la définition 6.20, reste à définir les deux nouveaux cas :

($\uparrow$) Partant d'un morphisme $f : A \xrightarrow{\bullet} B$, par curryfication on obtient un morphisme $g : \top \to B^A$. On construit $[\![\pi]\!]_{cl} = g; u_\parr^A; {s_{A,\perp,B}}^{-1} : \top \to \neg A \parr B$.

($\downarrow$) Partant d'un morphisme $f : \top \to A \parr B$, on obtient le morphisme $[\![\pi]\!]_{cl} : \perp^A \xrightarrow{\bullet} B$ par le lemme 14.21.

14.23. Proposition (Correction des catégories de contrôle linéaires pour MALLP)
*Soit $\mathbb{A}$ un catégorie de contrôle linéaire, si la preuve $\pi$ de* MALLP *se réduit en $\pi'$ alors* $[\![\pi]\!]_{cl} = [\![\pi']\!]_{cl}$.

Démonstration : Il s'agit de la même preuve que pour le théorème 6.21 en remplaçant les propriétés de CCC par celles de SMCC. On va se contenter de montrer que les conditions de cohérence, qui sont utilisées dans cette preuve, sont valides pour les catégories de contrôle linéaires.

Tout d'abord :

$$\begin{tikzcd}
& (\neg A \parr B) \parr C \arrow[dl, "s_{A,B,C}"'] \arrow[dr, "p_{A,B,D} \parr C"] & \\
\neg A \parr (B \parr C) \arrow[dr, "p_{A,B \parr C,D}"'] & & (\neg(D \parr A) \parr (D \parr B)) \parr C \arrow[dl, "s_{D \parr A, D \parr B, C}"] \\
& \neg(D \parr A) \parr (D \parr B \parr C) &
\end{tikzcd}$$

puisqu'il s'agit de la curryfication de :

$$\begin{tikzcd}
& (\neg A \parr B \parr C) \odot (A \parr D) \arrow[dl, "\delta_{\neg A \parr B \parr C, A, D}"'] \arrow[d, "\delta_{A \parr D, \neg A \parr B, C}"] \arrow[dr, "(p_{A,B,D} \parr C) \odot (A \parr D)"] & \\
((\neg A \parr B \parr C) \odot A) \parr D \arrow[d, "\delta_{A, \neg A \parr B, C} \parr D"'] & C \parr ((\neg A \parr B) \odot (A \parr D)) \arrow[dl, "C \parr \delta_{\neg A \parr B, A, D}"] \arrow[dr, "C \parr (p_{A,B,D} \odot (A \parr D))"] & (\neg(A \parr D) \parr B \parr D \parr C) \odot (A \parr D) \arrow[d, "\delta_{A \parr D, \neg(A \parr D) \parr B \parr D, C}"] \\
C \parr ((\neg A \parr B) \odot A) \parr D \arrow[dr, "C \parr ev \parr D"'] & & C \parr ((\neg(A \parr D) \parr B \parr D) \odot (A \parr D)) \arrow[dl, "C \parr (ev \odot (A \parr D))"] \\
& C \parr B \parr D &
\end{tikzcd}$$

les deux carrés du haut commutent par naturalité de $\delta_{A,B,C}$ et celui du bas par définition de $p_{A,B,C}$. De plus, $(\delta_{A,\neg A \parr B,C} \parr D); (C \parr ev \parr D) = ev \parr D$ ce qui permet de conclure.

On a également $id_{A^*}; p_{A,A,B} = id_{B \parr A^*}$ puisque le diagramme suivant commute :

$$\begin{tikzcd}
I \odot (B \parr A) \arrow[rr, "id_A{}^* \odot (B \parr A)"] \arrow[d, "\delta_{I,A,B}"'] & & A^A \odot (B \parr A) \arrow[d, "\delta_{A^A,A,B}"] \\
B \parr (I \odot A) \arrow[rr, "B \parr (id_A{}^* \odot A)"] \arrow[dr, "B \parr u_\odot"'] & & B \parr (A^A \odot B) \arrow[dl, "B \parr ev"] \\
& B \parr A &
\end{tikzcd}$$

par naturalité de $\delta_{A,B,C}$ et propriétés de SMCC. □

## 14.5 Logique classique affine

Si l'on continue à suivre la sémantique des jeux polarisés, en particulier à travers les questions de définissabilité, on est amené à ajouter la règle d'affaiblissement sur les formules négatives (celle-ci étant validée par les jeux) ce qui donne un autre sous-système de LLP que l'on appelle MAALP. Il est obtenu en ajoutant la règle suivante à MALLP :

$$\frac{\vdash \Gamma}{\vdash \Gamma, N}\uparrow w$$

On obtient, dans MAALP, la définissabilité des constantes multiplicatives à partir des constantes additives : $1 \simeq \downarrow\top$ et $\bot \simeq \uparrow 0$ (c'est la version linéaire des isomorphismes $1 \simeq !\top$ et $\bot \simeq ?0$ de LL que l'on retrouve dans LLP).

$$\dfrac{\dfrac{\dfrac{}{\vdash \top}\top}{\vdash \downarrow\top}\downarrow}{\vdash \downarrow\top, \bot}\bot \qquad\qquad \dfrac{\dfrac{}{\vdash 1}1}{\vdash 1, \uparrow 0}\uparrow w$$

14.24. Remarque : La question de savoir s'il faut considérer $I \simeq \top$ se pose également en logique classique affine avec quelques éléments de réponse supplémentaires. Puisque $\downarrow\top \simeq 1$, $I$ est l'élément neutre de $\odot$ et 1 celui de $\otimes$, on a :

$$\begin{aligned}
\downarrow\top &\simeq \downarrow(\top \odot I) \\
&\simeq \downarrow\top \otimes \downarrow I \\
&\simeq 1 \otimes \downarrow I \\
&\simeq \downarrow I
\end{aligned}$$

d'où, par réversibilité de $\downarrow$, $\downarrow\top \vdash I$, et puisque $\top$ est terminal $I \vdash \top$. Ce qui ne donne toujours pas pour autant $\top \simeq I$ !

14.25. Remarque : Si l'on considère, dans les espaces cohérents, la restriction de ! à $\downarrow$ en définissant $|\downarrow\mathcal{N}|$ comme l'ensemble des multi-ensembles d'éléments de $\mathcal{N}$ contenant au plus un élément (ce qui donne un espace cohérent isomorphe à $\bot \& \mathcal{N}$), on obtient un modèle de MAALP.

14.26. Proposition (Modèle de jeux pour MAALP)
*Les jeux polarisés fournissent un modèle de* MAALP.

Savoir lequel des deux systèmes MALLP et MAALP est le plus valable reste une question à étudier.

# Chapitre 15

# Dualité du calcul

La distinction entre *appel par nom* et *appel par valeur* est essentielle et bien connue en informatique. L'expression logique du lien entre ces deux modes de calcul sous forme d'une *dualité* (suggérée par Filinski [Fil89]) est réapparue récemment [Sel01, CH00] et semble être extrêmement riche. Il s'avère qu'elle s'exprime de manière particulièrement simple dans LLP comme la dualité entre positif et négatif : deux termes ou types sont duaux s'ils se traduisent de la même manière dans LLP, celui en appel par nom à l'aide de la traduction négative et celui en appel par valeur à l'aide de la traduction positive. Ce qui peut se résumer au niveau des formules par le tableau suivant :

| appel par nom | LLP | appel par valeur |
|---|---|---|
| négatif | $\overset{\perp}{\leftrightsquigarrow}$ | positif |
| $\neg A$ | $?A^\perp \leftrightsquigarrow\ !A^\perp$ | $\neg A$ |
| $A \wedge B$ | $A \mathbin{\&} B \leftrightsquigarrow A \oplus B$ | $A \vee B$ |
| $A \vee B$ | $A \parr B \leftrightsquigarrow A \otimes B$ | $A \wedge B$ |
| $V$ | $\top \leftrightsquigarrow 0$ | $F$ |
| $F$ | $\perp \leftrightsquigarrow 1$ | $V$ |
| $A \to B = \neg A \vee B$ | $!A \multimap B \leftrightsquigarrow B \otimes !A^\perp$ | $B - A = B \wedge \neg A$ |
| $A - B = \neg(A \to B)$ | $?(!A \otimes B^\perp) \leftrightsquigarrow\ !(B \multimap ?A)$ | $B \to A = \neg(B - A)$ |
| $\neg\neg_o A$ | $?A$ | $\neg_o \neg A$ |

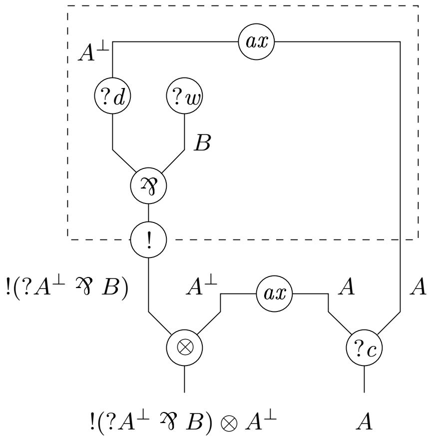
15.1. Remarque : Un point important de ce tableau est la définissabilité de $A - B$ en appel par nom et de $A \to B$ en appel par valeur, comme négation d'un autre connecteur. Ces connecteurs ne sont pas «primitifs» ce qui explique les difficultés que l'on a à les prendre en compte. Cette difficulté est surtout apparue historiquement dans le cas de l'implication par valeur car le connecteur $\to$ est a priori plus naturel d'un point de vue calculatoire que la différence. Ceci n'est semble-t-il pas fondé et entre parfaitement dans ce cadre de dualité.

15.2. Exemple
Les exemples 14.5 et 14.7 donnent des $\lambda\mu$-termes échangés par dualité puisque la traduction négative des premiers donne les mêmes réseaux que la traduction positive des seconds.

Comme nouvel exemple, on peut traduire le $\lambda\mu$-terme correspondant à call/cc en appel par nom :

$$\lambda f^{(A\to B)\to A}.\mu\alpha^A[\alpha](f)\lambda x^A.\mu\delta^B[\alpha]x : ((A\to B)\to A)\to A$$

ce qui donne, omettant le $\lambda f$, le réseau :




Ce réseau est la traduction positive du $\lambda$-terme $<\lambda d.x, x>$ (alors qu'il pourrait s'agir a priori d'un $\lambda\mu$-terme), ce qui donne le terme clos :

$$\lambda x^A.<\lambda d^B.x, x> : A\to((B\to A)\wedge A)$$

On retrouve bien ainsi les traductions syntaxiques de Selinger [Sel01] entre le $\lambda\mu$-calcul par nom et le $\lambda\mu$-calcul par valeur.

En ce qui concerne le lien avec les autres incarnations de cette dualité, on a vu que les formules négatives de LLP peuvent être interprétées par les objets d'une catégorie de contrôle (modèle de l'appel par nom) et que les formules positives peuvent être interprétées par les objets d'une catégorie de co-contrôle (modèle de l'appel par valeur).

Pour ce qui est du travail de Curien-Herbelin [CH00], les systèmes duaux $\bar{\lambda}\mu\tilde{\mu}_Q$-calcul et $\bar{\lambda}\mu\tilde{\mu}_T$-calcul qui correspondent à l'appel par valeur et à l'appel par nom sont traduits dans LLP par la traduction positive $(.)^+$ et la traduction négative $(.)^-$ en simulant la réduction. Ces traductions factorisent via LLP les CPS-traductions données par Curien-Herbelin de leur calcul dans le $\lambda$-calcul. Comme pour les traductions de Plotkin (chapitre 13), les deux CPS-traductions se trouvent unifiés par la traduction $(.)^b$ dans LLP.

Il ressort de cette étude que la dualité appel par nom/appel par valeur est une propriété clef de la logique classique. De plus les propriétés de cette dualité sont essentiellement indépendantes des règles structurelles (contraction et affaiblissement) et basées sur les questions de focalisation (décalages, boîtes, ...). L'analyse de cette dualité doit donc pouvoir être principalement effectuée dans le cadre de la *logique classique linéaire* (ou affine) : MALLP, MAALP, $\lambda\mu$-calcul linéaire, ludique, ...

# Annexes

# Annexe A

# Notations logiques

## A.1 Systèmes logiques

| | | |
|---|---|---|
| LL | Logique Linéaire | [Gir87] |
| MLL | fragment multiplicatif de LL | |
| MALL | fragment multiplicatif et additif de LL | |
| MELL | fragment multiplicatif et exponentiel de LL | |
| ILL | Logique Linéaire Intuitionniste | par exemple [Sch94] |
| taLL | sous-système de LL avec contextes additifs $?\Gamma$ | [DJS97] |
| $\mathsf{MALL}_{\mathrm{f}}$ | fragment multiplicatif et additif de la ludique | [Gir01] |
| LJ | Logique Intuitionniste | |
| LK | Logique Classique de Gentzen | [Gen67] |
| LC | Logique Classique de Girard | [Gir91a] |
| $\mathsf{LK}^{\mathrm{tq}}$ | Logique Classique de Danos-Joinet-Schellinx | [DJS97] |
| $\mathsf{LK}^{\eta}_{\mathrm{pol}}$ | sous-système polarisé et $\eta$-contraint de $\mathsf{LK}^{\mathrm{tq}}$ | [DJS97] |
| $\mathsf{LK}^{\eta,\rho}_{\mathrm{pol}}$ | sous-système $\rho$-contraint de $\mathsf{LK}^{\eta}_{\mathrm{pol}}$ | [QTdF96, LQTdF00] |
| LKQ | sous-système $\mathtt{q}$ avec $\rightarrow$ et $\forall$ de $\mathsf{LK}^{\mathrm{tq}}$ | [DJS95] |
| LKT | sous-système $\mathtt{t}$ avec $\rightarrow$ et $\forall$ de $\mathsf{LK}^{\mathrm{tq}}$ | [DJS95] |
| $\mathsf{LL}^{\flat}$ | raffinement de LL linéarisant la réduction additive | [Gir95a] |
| $\mathsf{MALL}^{\flat}$ | fragment multiplicatif et additif de $\mathsf{LL}^{\flat}$ | |
| LLP | Logique Linéaire Polarisée | chapitre 4 |
| $\mathsf{LL}_{\mathrm{pol}}$ | fragment polarisé de LL | définition 4.4 |
| $\mathsf{LL}^{+}_{\mathrm{pol}}$ | fragment polarisé de LL avec formules $?N$ et $!P$ | définition 4.6 |
| taLLP | sous-système de LLP avec contextes additifs négatifs | section 5.1.3 |
| MELLP | fragment multiplicatif et exponentiel de LLP | |
| MALLP | fragment multiplicatif et additif de LLP avec $\downarrow$ et $\uparrow$ | section 14.1 |
| MAALP | fragment multiplicatif et additif affine de LLP (MALLP + $\uparrow w$) | section 14.5 |

## A.2 Calcul des séquents

### A.2.1 Règles

Dans une règle du calcul des séquents introduisant le connecteur $\circledast$ :

$$\frac{\Gamma_1 \vdash A_1^i, \Delta_1 \qquad \dots \qquad \Gamma_n \vdash A_n^j, \Delta_n}{\Gamma_1, \dots, \Gamma_n \vdash \circledast(A_1^i, \dots, A_n^j), \Delta_1, \dots, \Delta_n}$$

on dit que la formule $\circledast(A_1^i, \ldots, A_n^j)$ est *principale* et que les formules $A_1^i$, ..., $A_n^j$ sont *actives*.

### A.2.2 $\eta$-expansion

Soit $\circledast$ un connecteur dont l'une des règles d'introduction (gauche ou droite) est réversible et l'autre non (c'est le cas de tous les connecteurs que l'on considère), une *$\eta$-expansion* de la preuve $\dfrac{}{A \circledast B \vdash A \circledast B}\,ax$ est une preuve obtenue à partir de $\dfrac{}{A \vdash A}\,ax$ (ou d'une $\eta$-expansion) et de $\dfrac{}{B \vdash B}\,ax$ (ou d'une $\eta$-expansion) en appliquant d'abord la règle d'introduction irréversible de $\circledast$ puis la règle réversible.

Cette notion s'applique de la manière naturelle si on est dans un calcul monolatère.

A.1. Exemple

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,ax}{\vdash A, ?A^\perp}\,?d}{\vdash !A, ?A^\perp}\,!}{\vdash !A \oplus (B^\perp \otimes C^\perp), ?A^\perp}\,\oplus_1 \qquad \dfrac{\dfrac{}{\vdash B^\perp \otimes C^\perp, B \parr C}\,ax}{\vdash !A \oplus (B^\perp \otimes C^\perp), B \parr C}\,\oplus_2}{\vdash !A \oplus (B^\perp \otimes C^\perp), ?A^\perp \,\&\, (B \parr C)}\,\&$$

# Annexe B

# LL

Le système LL a été introduit par J.-Y. Girard dans [Gir87].

## B.1 Formules

Les *formules linéaires* sont définies par la grammaire :

$$\begin{array}{rcccccccccccccccc} A & ::= & X & \mid & A \parr A & \mid & A \,\&\, A & \mid & \bot & \mid & \top & \mid & \forall X A & \mid & ?A \\ & \mid & X^\bot & \mid & A \otimes A & \mid & A \oplus A & \mid & 1 & \mid & 0 & \mid & \exists X A & \mid & !A \end{array}$$

Si $A$ est une formule, $A^\bot$ est définie en utilisant les lois de De Morgan par récurrence sur la taille de $A$ :

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

On a alors, pour toute formule $A$, $(A^\bot)^\bot = A$.

## B.2 Règles

$$\frac{}{\vdash A, A^\bot}\,ax \qquad\qquad \frac{\vdash \Gamma, A \qquad \vdash A^\bot, \Delta}{\vdash \Gamma, \Delta}\,cut$$

$$\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\,\parr \qquad\qquad \frac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}\,\otimes$$

$$\frac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \,\&\, B}\;\& \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B}\;\oplus_1 \qquad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B}\;\oplus_2$$

$$\frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}\;! \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}\;?d \qquad \frac{\vdash \Gamma}{\vdash \Gamma, ?A}\;?w \qquad \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}\;?c$$

$$\frac{}{\vdash \Gamma, \top}\;\top \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \bot}\;\bot \qquad \frac{}{\vdash 1}\;1$$

$$\frac{\vdash \Gamma, A}{\vdash \Gamma, \forall X A}\;\forall \qquad \frac{\vdash \Gamma, A[{}^{B}/_{X}]}{\vdash \Gamma, \exists X A}\;\exists$$

La variable $X$ n'est pas libre dans le contexte de la règle $\forall$.

# Annexe C

# $\mathsf{LK}^{(\mathrm{tq})}$

Le système $\mathsf{LK}^{\mathrm{tq}}$ a été introduit par Danos-Joinet-Schellinx dans [DJS97].

## C.1 Formules

Les formules de $\mathsf{LK}^{\mathrm{tq}}$ sont :

$$
\begin{array}{rcccccccccccccc}
A & ::= & X & \mid & A \wedge_m A & \mid & A \vee_a A & \mid & V & \mid & F & \mid & \exists X A \\
 & \mid & \neg X & \mid & A \vee_m A & \mid & A \wedge_a A & \mid & \neg V & \mid & \neg F & \mid & \forall X A
\end{array}
$$

et $\neg A$ est définie en appliquant les lois de De Morgan (comme pour $\mathsf{LL}$ à la section précédente) ce qui échange conjonction multiplicative (resp. additive) et disjonction multiplicative (resp. additive). On a alors $\neg\neg A = A$.

## C.2 Règles

$$
\frac{}{\vdash A, \neg A}\ ax \qquad \frac{\vdash \Gamma, A \qquad \vdash \neg A, \Delta}{\vdash \Gamma, \Delta}\ cut
$$

$$
\frac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \wedge_m B}\ \wedge_m \qquad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \vee_m B}\ \vee_m
$$

$$
\frac{\vdash \Gamma, A}{\vdash \Gamma, A \vee_a B}\ {\vee_a}^1 \qquad \frac{\vdash \Gamma, B}{\vdash \Gamma, A \vee_a B}\ {\vee_a}^2 \qquad \frac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \wedge_a B}\ \wedge_a
$$

$$
\frac{\vdash \Gamma}{\vdash \Gamma, A}\ wk \qquad \frac{\vdash \Gamma, A, A}{\vdash \Gamma, A}\ co
$$

$$
\frac{}{\vdash V}\ V \qquad \frac{\vdash \Gamma}{\vdash \Gamma, \neg V}\ \neg V \qquad \frac{}{\vdash \Gamma, \neg F}\ \neg F
$$

$$
\frac{\vdash \Gamma, A}{\vdash \Gamma, \forall X A}\ \forall \ \ X \text{ non libre dans } \Gamma \qquad \frac{\vdash \Gamma, A[{}^{B}/_{X}]}{\vdash \Gamma, \exists X A}\ \exists
$$

## C.3 $\mathtt{tq}$-protocole

À chaque formule (et ses sous-formules) d'une preuve de $\mathsf{LK}^{\mathtt{tq}}$, on associe une *couleur* $\mathtt{t}$ ou $\mathtt{q}$ permettant de déterminer comment les coupures doivent être éliminées (les couleurs doivent respecter les classes d'identité [DJS93]).

C.1. Définition (Coupures)
Une coupure est dite :
- de type $S_1$, si la formule coupée de couleur $\mathtt{q}$ n'est pas *principale*.
- de type $S_2$, si la formule coupée de couleur $\mathtt{q}$ est *principale* mais pas la formule de couleur $\mathtt{t}$.
- de type $L$, si les deux formules coupées sont *principales*.

Le $\mathtt{tq}$-protocole est défini selon le type des coupures : soit $\pi$ une preuve, on considère une coupure dans $\pi$ et on la réduit de la manière suivante :

($S_1$) La prémisse $\pi'$ de la formule $\mathtt{t}$ est dupliquée et placée aux feuilles de l'arbre structurel de la formule $\mathtt{q}$, trois cas sont possibles :
- la feuille est une règle logique et on introduit une coupure entre $\pi'$ et la conclusion de cette règle ;
- la feuille est un affaiblissement et on le remplace par des affaiblissements sur le contexte de la formule $\mathtt{t}$ ;
- la feuille est un axiome et on élimine la coupure axiome induite ce qui introduit $\pi'$ comme preuve de l'autre conclusion de l'axiome.

($S_2$) On duplique cette fois-ci la prémisse de la formule $\mathtt{q}$ de la même manière que dans le cas précédent.

($L$) On applique le cas clef correspondant.

Le $\mathtt{tq}$-protocole définit une élimination des coupures *confluente* et *fortement normalisante*.

C.2. Exemple
Une même preuve de $\mathsf{LK}$ peut être décorée de plusieurs manières en preuve de $\mathsf{LK}^{\mathtt{tq}}$, et ceci donne lieu à des réductions différentes :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash (\neg A)^{\mathtt{q}}, A^{\mathtt{t}}}\,ax}{\vdash (\neg A)^{\mathtt{q}}, A^{\mathtt{t}}, (A^{\mathtt{t}} \vee_a B^{\mathtt{t}})^{\mathtt{q}}}\,wk}{\vdash (\neg A)^{\mathtt{t}}, (A^{\mathtt{t}} \vee_a B^{\mathtt{t}})^{\mathtt{q}}, (A^{\mathtt{t}} \vee_a B^{\mathtt{t}})^{\mathtt{q}}}\,{\vee_a}^1}{\vdash (\neg A)^{\mathtt{t}}, (A^{\mathtt{t}} \vee_a B^{\mathtt{t}})^{\mathtt{q}}}\,co \qquad \dfrac{\dfrac{}{\vdash V^{\mathtt{t}}}\,V}{\vdash ((\neg A)^{\mathtt{q}} \wedge_a (\neg B)^{\mathtt{q}})^{\mathtt{t}}, V^{\mathtt{t}}}\,wk}{\vdash (\neg A)^{\mathtt{q}}, V^{\mathtt{t}}}\,cut$$

$$\downarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash V^{\mathtt{t}}}\,V}{\vdash (\neg A)^{\mathtt{q}}, V^{\mathtt{t}}}\,wk}{\vdash (\neg A)^{\mathtt{q}}, V^{\mathtt{t}}, V^{\mathtt{t}}}\,wk}{\vdash (\neg A)^{\mathtt{q}}, V^{\mathtt{t}}}\,co$$

et

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash (\neg A)^{\mathtt{t}}, A^{\mathtt{q}}}\ ax}{\vdash (\neg A)^{\mathtt{t}}, A^{\mathtt{q}}, (A^{\mathtt{q}} \vee_a B^{\mathtt{t}})^{\mathtt{t}}}\ wk}{\vdash (\neg A)^{\mathtt{t}}, (A^{\mathtt{q}} \vee_a B^{\mathtt{t}})^{\mathtt{t}}, (A^{\mathtt{q}} \vee_a B^{\mathtt{t}})^{\mathtt{t}}}\ {\vee_a}^1}{\vdash (\neg A)^{\mathtt{t}}, (A^{\mathtt{q}} \vee_a B^{\mathtt{t}})^{\mathtt{t}}}\ co \qquad \dfrac{\dfrac{}{\vdash V^{\mathtt{q}}}\ V}{\vdash ((\neg A)^{\mathtt{t}} \wedge_a (\neg B)^{\mathtt{q}})^{\mathtt{q}}, V^{\mathtt{q}}}\ wk}{\vdash (\neg A)^{\mathtt{t}}, V^{\mathtt{q}}}\ cut$$

$$\downarrow$$

$$\dfrac{\dfrac{}{\vdash V^{\mathtt{q}}}\ V}{\vdash (\neg A)^{\mathtt{t}}, V^{\mathtt{q}}}\ wk$$

En l'absence de couleurs ce protocole reste fortement normalisant [UB99].

# Annexe D

# $\lambda\mu$-calcul

Le $\lambda\mu$-*calcul* a été introduit par M. Parigot dans [Par92].

## D.1 Termes

Les $\lambda\mu$-*termes* sont construits à partir de deux ensembles disjoints de variables, les $\lambda$-*variables* notées $x$, $y$, $z$, ... et les $\mu$-*variables* notées $\alpha$, $\beta$, $\gamma$, ...

$$t \quad ::= \quad x \quad | \quad \lambda x.t \quad | \quad (t)t \quad | \quad \mu\beta[\alpha]t$$

$\lambda$ et $\mu$ sont des lieurs et les $\lambda\mu$-termes seront toujours considérés modulo $\alpha$-équivalence. On note $x \in t$ (resp. $\alpha \in t$) si la variable $x$ (resp. $\alpha$) est libre dans $t$.

Si $t$ est un terme et $\alpha$ une $\mu$-variable, on dit que $[\alpha]t$ est un *terme nommé*.

D.1. Définition (Variable de tête)
On définit la *variable de tête* d'un terme par récurrence sur la taille du terme :
- la variable de tête de $x$ est $x$,
- la variable de tête de $\lambda x.t$ est celle de $t$,
- la variable de tête de $(t)u$ est celle de $t$,
- la variable de tête de $\mu\beta[\alpha]t$ est celle de $t$.

## D.2 Typage

Les types sont $A \quad ::= \quad X \quad | \quad A \to A$ , les jugements de typage sont de la forme $\Gamma \vdash t : A \mid \Delta$.
Les règles de typage :

$$\frac{}{x : A \vdash x : A \mid} \text{var}$$

$$\frac{\Gamma, x : A \vdash t : B \mid \Delta}{\Gamma \vdash \lambda x.t : A \to B \mid \Delta} \text{abs}$$

$$\frac{\Gamma \vdash u : A \to B \mid \Delta \qquad \Gamma' \vdash v : A \mid \Delta'}{\Gamma, \Gamma' \vdash (u)v : B \mid \Delta, \Delta'} \text{app}$$

$$\frac{\Gamma \vdash t : A \mid \beta : B, \Delta}{\Gamma \vdash \mu\beta[\alpha]t : B \mid \alpha : A, \Delta} \mu$$

où $\Gamma, \Gamma'$ dénote l'union des deux ensembles de déclarations $\Gamma$ et $\Gamma'$ avec la convention qu'ils ne doivent pas contenir deux déclarations contradictoires comme $x : A$ et $x : B$ avec $A \neq B$.

D.2. Lemme (Contexte de typage)
*Si le jugement $\Gamma \vdash t : A \mid \Delta$ est dérivable alors $\Gamma$ et $\Delta$ contiennent, respectivement, les $\lambda$-variables et les $\mu$-variables libres de $t$.*

Démonstration : Par récurrence sur la taille de la dérivation de typage, en considérant chaque cas de dernière règle :

(var) Immédiat.

(abs) $\lambda x.t$ ne contient plus la $\lambda$-variable $x$ libre, et elle a été supprimée du contexte $\Gamma$.

(app) Les variables libres de $(u)v$ sont la réunion de celles de $u$ et de celles de $v$.

($\mu$) $\mu\beta[\alpha]t$ ne contient plus la $\mu$-variable $\beta$ libre, et elle a été supprimée du contexte $\Delta$ mais contient désormais la $\mu$-variable libre $\alpha$ qui a été rajoutée dans $\Delta$. □

D.3. Lemme (Unicité du typage)
*Étant donnés un $\lambda\mu$-terme $t$, un type $A$ et des types $A_1$, ..., $A_n$ pour ses variables libres, il existe au plus une dérivation de typage pour $t$ lui donnant le type $A$ et donnant les types $A_1$, ..., $A_n$ à ses variables libres.*

Démonstration : Par récurrence sur la taille de $t$ en considérant chacune des constructions de termes. D'après le lemme D.2, on sait déjà qu'il existe au plus un jugement de typage.

($x$) Immédiat.

($\lambda$) La dernière règle de typage de $\lambda x.t$ ne peut être que (abs) et on applique l'hypothèse de récurrence à $t$.

(@) La dernière règle de typage de $(u)v$ ne peut être que (app) et on applique l'hypothèse de récurrence à $u$ et $v$.

($\mu$) La dernière règle de typage de $\mu\beta[\alpha]t$ ne peut être que ($\mu$) et on applique l'hypothèse de récurrence à $t$. □

D.4. Remarque : On a utilisé le fait que le système de typage est *dirigé par la syntaxe*, *i.e.* étant donné un terme il n'existe qu'une règle de typage qui a pu y mener.

En particulier si $t$ est un $\lambda\mu$-terme clos, étant donné un type $A$, il existe au plus une dérivation de typage pour typer $t$ de type $A$.

## D.3 Réduction

### D.3.1 Appel par nom

On reprend les règles de réduction telles qu'elles ont été données par M. Parigot [Par92] pour l'*appel par nom*.

$$\begin{array}{rcll}
(\lambda x.u)v & \to_\beta & u[^v/_x] & \\
(\mu\alpha.u)v & \to_\mu & \mu\alpha.u[^{[\alpha](w)v}/_{[\alpha]w}] & \\
 & & & \\
\lambda x.(t)x & \to_\eta & t & x \notin t \\
 & & & \\
[\beta]\mu\alpha.t & \to_\rho & t[^\beta/_\alpha] & \\
\mu\alpha[\alpha]t & \to_\theta & t & \alpha \notin t
\end{array}$$

On dit que $u$ se réduit en $v$ (noté $u \to v$) si $u$ se réduit en $v$ par une $\beta$- ou $\mu$-réduction.

### D.3.2 Appel par valeur

La variante en *appel par valeur* du $\lambda\mu$-calcul a été donnée par L. Ong et C. Stewart [OS97]. Les règles de réduction sont les suivantes :

$$\begin{array}{rll} (\lambda x.t)v & \to_{\beta_v} & t[{}^{v}/_{x}] \\ (\mu\alpha.t)t' & \to_{\mu} & \mu\alpha.t[{}^{[\alpha](u)t'}/_{[\alpha]u}] \\ (v)\mu\alpha.t & \to_{\mu_v} & \mu\alpha.t[{}^{[\alpha](v)u}/_{[\alpha]u}] \\ \\ [\beta]\mu\alpha.t & \to_{\rho} & t[{}^{\beta}/_{\alpha}] \\ \mu\alpha[\alpha]t & \to_{\theta} & t \qquad\qquad\qquad \alpha \notin t \end{array}$$

où $v$ est une *valeur*, c'est-à-dire commence par un $\lambda$.

## D.4 Machine de Krivine

La machine de Krivine [Kri92] est une machine abstraite à pile permettant d'évaluer les $\lambda\mu$-termes. Nous rappelons ici la version en appel par nom.

Un état de la machine est une paire $c, \pi$ où $c$ est une clôture, c'est-à-dire une paire $(t, e)$. $t$ est un $\lambda\mu$-terme, $\pi$ est une pile de clôtures et $e$ est un environnement c'est-à-dire une pile de paires $(x = c)$ où $x$ est une $\lambda$-variable et $(\alpha = \pi)$ où $\alpha$ est une $\mu$-variable.

Les transitions de la machine sont déterminées par le terme de la clôture de la manière suivante :

$$\frac{(x, e), \pi}{c, \pi}\ (\texttt{jump})$$

$$\frac{(\lambda x.u, e), c :: \pi}{(u, (x = c) :: e), \pi}\ (\texttt{pop}) \qquad \frac{((u)v, e), \pi}{(u, e), (v, e) :: \pi}\ (\texttt{push})$$

$$\frac{(\mu\alpha.u, e), \pi}{(u, (\alpha = \pi) :: e), \varepsilon}\ (\texttt{store}) \qquad \frac{([\alpha]u, e), \pi}{(u, e), \pi'}\ (\texttt{restore})$$

Dans (`jump`), on dépile dans $e$ jusqu'à trouver une affectation $(x = c)$ pour $x$.

Dans (`restore`), on dépile dans $e$ jusqu'à trouver une affectation $(\alpha = \pi')$ pour $\alpha$.

## D.5 Extension aux autres connecteurs

On a décrit la partie propositionnelle du $\lambda\mu$-calcul tel que l'a définie M. Parigot avec $\to$ comme seul connecteur. Il est possible d'étendre ce calcul aux autres connecteurs classiques et au second ordre, on parlera de $\lambda\mu$-calcul complet. En regroupant par exemple les calculs de Parigot [Par92] avec second ordre et de Selinger [Sel01], on obtient :

Les termes et les types :

$$\begin{array}{ccccccccccccc} t & ::= & x & \mid & \star & \mid & \lambda x.t & \mid & (t)t & \mid & \mu\alpha.t & \mid & [\alpha]t \\ & \mid & <t, t> & \mid & \pi_1 t & \mid & \pi_2 t & \mid & \mu(\alpha, \beta).t & \mid & [\alpha, \beta]t & & \end{array}$$

$$A \ ::= \ X \mid A \to A \mid A \wedge A \mid A \vee A \mid V \mid F \mid \forall X A$$

Et les règles de typage :

$$\frac{}{x : A \vdash x : A \mid}\ \text{var} \qquad \frac{}{\Gamma \vdash \star : V \mid \Delta}\ \text{const}$$

$$\frac{\Gamma \vdash t : A \mid \Delta}{\Gamma \vdash [\alpha]t : F \mid \alpha : A, \Delta}\ \text{name} \qquad \frac{\Gamma \vdash t : F \mid \alpha : A, \Delta}{\Gamma \vdash \mu\alpha.t : A \mid \Delta}\ \mu - \text{abs}$$

$$\frac{\Gamma, x : A \vdash t : B \mid \Delta}{\Gamma \vdash \lambda x.t : A \to B \mid \Delta}\ \text{abs} \qquad \frac{\Gamma \vdash u : A \to B \mid \Delta \qquad \Gamma' \vdash v : A \mid \Delta'}{\Gamma, \Gamma' \vdash (u)v : B \mid \Delta, \Delta'}\ \text{app}$$

$$\frac{\Gamma \vdash u : A \mid \Delta \qquad \Gamma \vdash v : B \mid \Delta}{\Gamma \vdash <u, v> : A \wedge B \mid \Delta}\ \text{paire}$$

$$\frac{\Gamma \vdash t : A \wedge B \mid \Delta}{\Gamma \vdash \pi_1 t : A \mid \Delta}\ \text{proj}_1 \qquad \frac{\Gamma \vdash t : A \wedge B \mid \Delta}{\Gamma \vdash \pi_2 t : B \mid \Delta}\ \text{proj}_2$$

$$\frac{\Gamma \vdash t : F \mid \alpha : A, \beta : B, \Delta}{\Gamma \vdash \mu(\alpha, \beta).t : A \vee B \mid \Delta}\ \mu - \text{abs}_d \qquad \frac{\Gamma \vdash t : A \vee B \mid \Delta}{\Gamma \vdash [\alpha, \beta]t : F \mid \alpha : A, \beta : B, \Delta}\ \text{name}_d$$

$$\frac{\Gamma \vdash t : A \mid \Delta}{\Gamma \vdash t : \forall X A \mid \Delta}\ \text{gen} \qquad \frac{\Gamma \vdash t : \forall X A \mid \Delta}{\Gamma \vdash t : A[{}^{B}/_{X}] \mid \Delta}\ \text{inst}$$

La variable $X$ n'est pas libre dans le contexte de la règle (gen).

# Bibliographie


[AJ94] Samson Abramsky and Radha Jagadeesan. New foundations for the geometry of interaction. *Information and Computation*, 111(1) :53–119, 1994.

[AJM00] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Information and Computation*, 163(2) :409–470, December 2000.

[And90] Jean-Marc Andreoli. *Proposition pour une synthèse des paradigmes de la programmation logique et de la programmation par objets.* Thèse de doctorat, Université Paris VI, June 1990.

[Bai99] Patrick Baillot. *Approches dynamiques en sémantique de la logique linéaire : jeux et géométrie de l'interaction.* Thèse de doctorat, Université Aix-Marseille II, January 1999.

[BCST96] Richard Blute, Robin Cockett, Robert Seely, and Todd Trimble. Natural deduction and coherence for weakly distributive categories. *Journal of Pure and Applied Algebra*, 113 :229–296, 1996.

[CH00] Pierre-Louis Curien and Hugo Herbelin. The duality of computation. In *Proceedings of the International Conference on Functional Programming*, September 2000.

[CS97] Robin Cockett and Robert Seely. Weakly distributive categories. *Journal of Pure and Applied Algebra*, 114 :133–173, 1997.

[Dan90] Vincent Danos. *La Logique Linéaire appliquée à l'étude de divers processus de normalisation (principalement du $\lambda$-calcul).* Thèse de doctorat, Université Paris VII, 1990.

[DHR96] Vincent Danos, Hugo Herbelin, and Laurent Regnier. Games semantics and abstract machines. In *Proceedings of the eleventh annual IEEE symposium on Logic In Computer Science*. IEEE Computer Society Press, 1996.

[DJS93] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. The structure of exponentials : uncovering the dynamics of linear logic proofs. In G. Gottlob, A. Leitsch, and D. Mundici, editors, *Computational Logic and Proof Theory*, volume 713 of *Lecture Notes in Computer Science*, pages 159–171. Springer, August 1993.

[DJS95] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. LKQ and LKT : Sequent calculi for second order logic based upon dual linear decompositions of classical implication. In Girard et al. [GLR95], pages 211–224.

[DJS97] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. A new deconstructive logic : linear logic. *Journal of Symbolic Logic*, 62(3) :755–807, September 1997.

[DK00] Vincent Danos and Jean-Louis Krivine. Disjunctive tautologies as synchronisation schemes. In Peter Clote and Helmut Schwichtenberg, editors, *Computer Science Logic*, volume 1862 of *Lecture Notes in Computer Science*, pages 292–301. Springer, August 2000.

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

[DR95] Vincent Danos and Laurent Regnier. Proof-nets and the Hilbert space. In Girard et al. [GLR95], pages 307–328.

[DR99] Vincent Danos and Laurent Regnier. Reversible, irreversible and optimal $\lambda$-machines. *Theoretical Computer Science*, 227 :79–97, September 1999.

[Fel86] Mathias Felleisen. *The calculi of $\lambda_v$-CS conversion : A syntactic theory of control and state in imperative higher order programming languages.* Ph.D. thesis, Indiana University, 1986.

[Fil89] Andrzej Filinski. Declarative continuations and categorical duality. Master's thesis, computer science department, University of Copenhagen, August 1989. DIKU Report 89/11.

[Gen67] Gerhard Gentzen. *The Collected Works of G. Gentzen.* North-Holland, 1967.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50 :1–102, 1987.

[Gir88] Jean-Yves Girard. Geometry of interaction I : an interpretation of system $F$. In Ferro, Bonotto, Valentini, and Zanardo, editors, *Logic Colloquium '88.* North-Holland, 1988.

[Gir91a] Jean-Yves Girard. A new constructive logic : classical logic. *Mathematical Structures in Computer Science*, 1(3) :255–296, 1991.

[Gir91b] Jean-Yves Girard. Quantifiers in linear logic II. In Corsi and Sambin, editors, *Nuovi problemi della logica e della filosofia della scienza*, pages 79–90, Bologna, 1991. CLUEB.

[Gir95a] Jean-Yves Girard. Geometry of interaction III : accommodating the additives. In Girard et al. [GLR95], pages 329–389.

[Gir95b] Jean-Yves Girard. Linear logic : its syntax and semantics. In Girard et al. [GLR95], pages 1–42.

[Gir96] Jean-Yves Girard. Proof-nets : the parallel syntax for proof-theory. In Ursini and Agliano, editors, *Logic and Algebra*, New York, 1996. Marcel Dekker.

[Gir99] Jean-Yves Girard, editor. *Typed Lambda Calculi and Applications '99*, volume 1581 of *Lecture Notes in Computer Science.* Springer, April 1999.

[Gir01] Jean-Yves Girard. Locus solum : From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(3) :301–506, June 2001.

[GLR95] Jean-Yves Girard, Yves Lafont, and Laurent Regnier, editors. *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Note Series.* Cambridge University Press, 1995.

[GSS92] Jean-Yves Girard, André Scedrov, and Phil Scott. Normal forms and cut-free proofs as natural transformations. In Y. Moschovakis, editor, *Logic from Computer Science*, volume 21, pages 217–242. MSRI Publications, Springer, 1992.

[Gue99] Stefano Guerrini. Correctness of multiplicative proof nets is linear. In *Proceedings of the fourteenth annual IEEE symposium on Logic In Computer Science*, pages 454–463, Trento, July 1999. IEEE Computer Society Press.

[Har99] Russel Harmer. *Games and Full Abstraction for Nondeterministic Languages.* Ph.D. thesis, Imperial College and University of London, 1999.

[HD97] John Hatcliff and Olivier Danvy. Thunks and the $\lambda$-calculus. *Journal of Functional Programming*, 7(3) :303–319, May 1997.

[HO00] Martin Hyland and Luke Ong. On full abstraction for PCF. *Information and Computation*, 163(2) :285–408, December 2000.

[HY99] Kohei Honda and Nobuko Yoshida. Game-theoretic analysis of call-by-value computation. *Theoretical Computer Science*, 221 :393–456, 1999.

[Kle52] Stephen Kleene. *Introduction to Metamathematics*. North-Holland, 1952.

[Kri90] Jean-Louis Krivine. *Lambda-Calcul : Types et Modèles*. Etudes et Recherches en Informatique. Masson, 1990.

[Kri92] Jean-Louis Krivine. Un interpréteur du lambda-calcul. Notes non publiées, 1992.

[Kri94] Jean-Louis Krivine. Classical logic, storage operators and second order $\lambda$-calculus. *Annals of Pure and Applied Logic*, 68 :53–78, 1994.

[Lam94] François Lamarche. Proof nets for intuitionistic linear logic I : Essential nets. Preliminary report, April 1994.

[Lam95] François Lamarche. Games semantics for full propositional linear logic. In *Proceedings of the tenth annual IEEE symposium on Logic In Computer Science*. IEEE Computer Society Press, 1995.

[Lau99] Olivier Laurent. Polarized proof-nets : proof-nets for LC (extended abstract). In Girard [Gir99], pages 213–227.

[Lau01] Olivier Laurent. A token machine for full geometry of interaction (extended abstract). In Samson Abramsky, editor, *Typed Lambda Calculi and Applications '01*, volume 2044 of *Lecture Notes in Computer Science*, pages 283–297. Springer, May 2001.

[Lau02a] Olivier Laurent. Polarized games (extended abstract). In *Proceedings of the seventeenth annual IEEE symposium on Logic In Computer Science*. IEEE Computer Society Press, July 2002.

[Lau02b] Olivier Laurent. Polarized proof-nets and $\lambda\mu$-calculus. To appear in *Theoretical Computer Science*, 2002.

[LQTdF00] Olivier Laurent, Myriam Quatrini, and Lorenzo Tortora de Falco. Polarized and focalized linear and classical proofs. Prépublication 24, Institut de Mathématiques de Luminy, Marseille, France, September 2000. To appear in *Annals of Pure and Applied Logic*.

[LR02] Olivier Laurent and Laurent Regnier. A linear analysis of CPS-translations. En préparation, 2002.

[LTdF01] Olivier Laurent and Lorenzo Tortora de Falco. Slicing polarized additive normalization. Quaderno 12, Istituto per le Applicazioni del Calcolo (CNR), Roma, Italia, March 2001. To appear in the TMR LINEAR Book on Linear Logic.

[McC96] Guy McCusker. *Games and Full Abstraction for a Functional Metalanguage with Recursive Types*. Ph.D. thesis, Imperial College and University of London, 1996. Published in Springer-Verlag's Distinguished Dissertations in Computer Science series, 1998.

[Mét94] François Métayer. Homology of proofnets. *Archive for Mathematical Logic*, 33 :169–188, 1994.

[Oka99] Mitsuhiro Okada. Phase semantic cut elimination and normalization proofs of first- and higher-order linear logic. *Theoretical Computer Science*, 227 :333–396, September 1999.

[OS97] Luke Ong and Charles Stewart. A Curry-Howard foundation for functional computation with control. In *Proceedings of the $24^{th}$ Annual ACM Symposium on Principles of Programming Languages*, pages 215–227. Association for Computing Machinery, ACM Press, January 1997.

[Par92] Michel Parigot. $\lambda\mu$-calculus : an algorithmic interpretation of classical natural deduction. In *Proceedings of Internationnal Conference on Logic Programming and Automated Deduction*, volume 624 of *Lecture Notes in Computer Science*, pages 190–201. Springer, 1992.

[Plo75] Gordon Plotkin. Call-by-name, call-by-value and the $\lambda$-calculus. *Theoretical Computer Science*, 1 :125–159, 1975.

[PR97] John Power and Edmund Robinson. Premonoidal categories and notions of computation. *Mathematical Structures in Computer Science*, 7(5) :453–468, October 1997.

[QTdF96] Myriam Quatrini and Lorenzo Tortora de Falco. Polarisation des preuves classiques et renversement. *Compte-Rendu de l'Académie des Sciences de Paris*, 323 :113–116, 1996.

[Qua96] Myriam Quatrini. A denotational semantics of LC2. *Archive for Mathematical Logic*, 35 :1–32, 1996.

[Reg92] Laurent Regnier. *Lambda-Calcul et Réseaux.* Thèse de doctorat, Université Paris VII, 1992.

[Reg94] Laurent Regnier. Une équivalence sur les lambda-termes. *Theoretical Computer Science*, 126 :281–292, 1994.

[Sch94] Harold Schellinx. *The noble art of linear decorating.* ILLC dissertation series, Universiteit van Amsterdam, February 1994.

[Sel01] Peter Selinger. Control categories and duality : on the categorical semantics of the lambda-mu calculus. *Mathematical Structures in Computer Science*, 11(2) :207–260, April 2001.

[TdF00] Lorenzo Tortora de Falco. *Réseaux, cohérence et expériences obsessionnelles.* Thèse de doctorat, Université Paris VII, January 2000.

[TdF01] Lorenzo Tortora de Falco. Obsessional experiments for linear logic proof-nets. To appear in *Mathematical Structures in Computer Science*, January 2001.

[UB99] Christian Urban and Gavin Bierman. Strong normalisation of cut-elimination in classical logic. In Girard [Gir99], pages 365–380.

# Index

**Résumé :**

Issue des travaux sur la logique linéaire et l'analyse calculatoire de la logique classique, la notion de polarités semble jouer un rôle essentiel dans l'étude actuelle des systèmes logiques. La polarisation est une contrainte qui simplifie les objets tout en conservant une expressivité suffisante d'un point de vue informatique.

L'objet de cette thèse est d'étudier et d'exploiter cette nouvelle structure afin en particulier de mettre à jour les relations entre la logique classique et la logique linéaire (LL). L'introduction des polarités dans LL permet de mieux appréhender ce vaste système et de prolonger le développement de différents outils trop complexes en l'absence de cette contrainte. Nous définissons ainsi, pour la logique linéaire polarisée (LLP), des réseaux de preuve intégrant les connecteurs additifs de manière satisfaisante, une sémantique des jeux polarisés qui réconcilie jeux et dualité, une géométrie de l'interaction parallèle et d'autres sémantiques dénotationnelles basées sur des notions connues (espaces de corrélation, catégories de contrôle).

Il est important de montrer que malgré cette contrainte, LLP reste un système suffisamment expressif. Pour cela nous étudions en détail les traductions des différents systèmes de logique classique déterministe connus (LC, $\lambda\mu$-calcul, ...) aussi bien en appel par nom qu'en appel par valeur. De surcroît, les traductions obtenues pour ces systèmes sont plus simples que celles vers LL.

Enfin la souplesse de ces traductions nous permet d'analyser plus finement certaines propriétés de la logique classique tout comme LL permet d'analyser la logique intuitionniste. On peut ainsi étudier un équivalent linéaire des CPS-traductions.

**Mots-Clés :** Logique linéaire – Logique classique – Lambda calcul – Polarités – Réseaux de preuve – Sémantique des jeux – Géométrie de l'interaction.

**Title :** A study of polarization in logic

**Abstract :**

Coming from the study of linear logic and from the computational analysis of classical logic, the notion of polarities seems to have a very important place in the recent study of logical systems. The polarization constraint simplifies objects without reducing the computational expressivity too much.

This thesis deals with the study of this new structure given by polarities in order to enlighten the relations between classical logic and linear logic (LL). The introduction of polarities in LL allows to better understand this complex system. In particular, we define for this polarized linear logic (LLP) : a notion of proof-nets dealing with the additive connectives, a polarized game semantics reconciliating games and duality, a parallel geometry of interaction and some other denotational semantics based on already known structures (correlation spaces, control categories).

An important point is that LLP is still expressive enough. We precisely study the translations of the various deterministic systems for classical logic (LC, $\lambda\mu$-calculus, ...) both for call-by-name evaluation and for call-by-value. Moreover these translations are simpler than translations of the classical systems into LL.

These simplified translations allow to precisely analyze in LLP some properties of classical logic as LL allows to analyze intuitionistic logic. In particular it is possible to study a linear equivalent of CPS-translations.

**Keywords :** Linear logic – Classical logic – Lambda calculus – Polarities – Proof-nets – Game semantics – Geometry of interaction.

**Discipline :** Mathématiques

**Laboratoire :** Institut de Mathématiques de Luminy
163, avenue de Luminy - case 907 - 13288 Marseille cedex 09