



THÈSE

pour l'obtention du diplôme de

DOCTEUR DE L'UNIVERSITÉ PARIS DIDEROT

Spécialité INFORMATIQUE

LOGIQUE ET INTERACTION :
UNE ÉTUDE SÉMANTIQUE DE LA TOTALITÉ

Présentée et soutenue publiquement par

Pierre CLAIRAMBAULT

le 19 Février 2010, devant le jury composé de

Antonio	BUCCIARELLI	
Thierry	COQUAND	Rapporteur
Pierre-Louis	CURIEN	
Dan	GHICA	
Russ	HARMER	Directeur
Martin	HYLAND	Rapporteur
Luigi	SANTOCANALE	

Remerciements

Cette thèse a duré plus de trois ans, et de nombreuses personnes y ont contribué de près ou de loin, que ce soit par un soutien actif, par des discussions scientifiques, ou juste par leur présence.

Tout d'abord, je voudrais remercier Russ Harmer pour son encadrement, pour ses précieux conseils et pour m'avoir fait partager son expertise de la sémantique de jeux (et non pas «sémantique des jeux» !). Les moments passés à discuter ensemble ont été parmi les plus agréables de cette thèse. Merci également à Pierre-Louis Curien, pour avoir accepté d'être mon directeur officiel, et pour ses précieux conseils en fin de parcours.

Merci à Thierry Coquand et Martin Hyland pour l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs de cette thèse, pour leur lecture attentive de mon manuscrit, et pour leur précieux point de vue sur mon travail. Merci également à Antonio Bucciarelli, Dan Ghica et Luigi Santocanale pour avoir bien voulu participer au jury.

Une pensée pour mes camarades de rédaction Stéphane Gimenez, Grégoire Henry et Christine Tasson, grâce à qui je n'étais pas seul dans les couloirs de Chevaleret en plein mois d'août, lorsque le soleil se fait nettement plus attirant que la lumière blafarde des néons. Merci particulièrement à Stéphane et Grégoire pour cette semaine là à la campagne, qui me fut d'un grand réconfort alors que μLJ refusait de normaliser.

Je tiens aussi à remercier Julien Cristau, Jonas Frey, Florian Horn, Emmanuel Jeandel, Fabien Renaud, Nicolas Tabareau et Stéphane Zimmermann pour avoir relu des versions préliminaires de cette thèse, et contribué par leurs commentaires et leurs remarques à l'améliorer considérablement.

De nombreux chercheurs d'ici ou d'ailleurs ont participé à mes travaux, que ce soit par des discussions, des commentaires ou de simples encouragements. Je voudrais remercier Peter Dybjer pour nos discussions sur les jeux et les catégories pour les types dépendants et pour son accueil chaleureux à Chalmers, Laurent Regnier pour nos discussions sur la PAM et son accueil à l'IML, Olivier Laurent pour nos discussions sur les problèmes de normalisation en sémantique de jeux, Pierre Hyvernats pour ses commentaires précieux sur mon travail, Paul-André Melliès pour des discussions sur de nombreux sujets allant des modèles de jeux de la logique linéaire aux modèles catégoriques des types dépendants et Luigi Santocanale pour m'avoir communiqué sa passion pour les points fixes. Merci à Robin Cockett pour m'avoir encouragé à poursuivre mes investigations sur mon modèle de jeux des points fixes. I will find the free μ -closed category!

Je ne remercie *pas* les non-fenêtres de Chevaleret.

Cette thèse doit beaucoup au laboratoire PPS et à son atmosphère chaleureuse. Un grand merci à Odile et Audrey pour leur présence et leur efficacité. Je remercie également tous mes compagnons de thèse, dans un ordre ressemblant à l'ordre chronologique mais pas tout à fait : Joachim, Samuel M, Nicolas, Caroline, Séverine, Grégoire, Manu, Samuel H, Fabien

T, Florian, Sylvain, Marie, Christine, Barbara, Fabien R, Mehdi, Gabriel, Thibaut, Jonas, Stéphane {Gi, Gl, Z}, Antoine, Alexis, Guillaume... Merci aussi à Joseph, Thomas et Yann pour de passionnantes parties de tennis dans tous les courts du 13^e arrondissement. Merci également à tous les vieux (et moins vieux) de l'ENS Lyon qui fréquentent #sos. J'y laisserai sûrement trainer encore longtemps un **irssi** dans un **screen**. Enfin, merci à tous ceux que j'ai oublié.

Merci à mes parents Alix et François, qui ont toujours été là pour moi, même lorsque je le leur rendais bien peu. Merci à ma sœur Christine et à mes frères Jean-Marie et Pascal, ainsi qu'au cousin Nicolas (whitenoterocks). Merci à mes amis, qui ne m'ont jamais lâché dans les moments difficiles.

Enfin, merci à Anaïs. J'ai beau faire de la sémantique, c'est grâce à ta présence que je réalise que tout cela a du sens.

À Mamet.

Table des matières

Introduction	11
1 Théorie des catégories	19
1.1 Catégories	20
1.1.1 Définitions basiques	20
1.1.2 Foncteurs, transformations naturelles	22
1.2 Structure catégorique du calcul	23
1.2.1 Structure bicartésienne	23
1.2.2 Catégories cartésiennes fermées	25
1.2.3 Coproduits faibles fonctoriels	26
1.3 Préliminaires spécifiques	27
1.3.1 Algèbres initiales et coalgèbres finales	27
1.3.2 Catégories de co-Kleisli et relèvements	29
2 Sémantique de Jeux	33
2.1 Jeux et stratégies	35
2.1.1 Coups, arènes	35
2.1.2 Constructions sur les arènes	37
2.1.3 Parties légales	39
2.1.4 Stratégies et composition	40
2.2 Une catégorie cartésienne fermée	44
2.2.1 Fils et stratégies filaires	44
2.2.2 Vue, visibilité	46
2.2.3 Correspondance de Galois et saturation	50
2.2.4 Innocence	51
2.2.5 Structure cartésienne fermée	53
2.2.6 Coproduits faibles fonctoriels	54
2.3 Vers la complétude	57
2.3.1 Bon parenthésage	57
2.3.2 Définissabilité	58
2.4 Préliminaires spécifiques	60
2.4.1 Types récursifs à la McCusker	60
2.4.2 Isomorphismes en sémantique de jeux	61

I	Interactions et Totalité	65
3	λ-calcul et normalisation	67
3.1	λ -calcul unaire, définition et normalisation	68
3.2	Réduction linéaire de tête	72
3.2.1	Définitions	73
3.2.2	Propriétés	76
3.2.3	Normalisation	77
3.3	Machines abstraites	79
3.3.1	Réduction dirigée	79
3.3.2	Machine de Krivine	80
3.3.3	Machine à pointeurs	82
4	Pointeurs et Normalisation	93
4.1	Structures de pointeurs	94
4.1.1	Arènes pures et écrasement	94
4.1.2	Définitions et propriétés	94
4.2	Théorèmes de finitude	103
4.2.1	PAM inversée et réduction linéaire de tête	103
4.2.2	Agents, interactions et réalisabilité	106
4.2.3	Approche directe	115
4.3	Conséquences	120
4.3.1	Résultats intermédiaires	120
4.3.2	Stratégies finies, bornées, noethériennes	123
II	Points fixes en Sémantique des Jeux	129
5	Une logique avec points fixes : μLJ	131
5.1	Règles	132
5.1.1	Formules, séquents	132
5.1.2	Fragments	135
5.2	Réduction	136
5.2.1	Règles de réduction	136
5.2.2	Coupures irréductibles	146
5.3	Expressivité	148
5.3.1	Quotient structurel	148
5.3.2	Rappel : définition du système T	149
5.3.3	Traduction dans μLJ	150
6	Sémantique catégorique de μLJ	153
6.1	Catégories faiblement μ -fermées	153
6.2	Relèvement à \mathcal{C}_Γ	155
6.2.1	Foncteurs forts	155
6.2.2	Cas contravariant	157
6.2.3	Relèvement des points fixes	159
6.2.4	Uniformité du relèvement	160

6.2.5	Discussion : construction des forces	164
6.3	Modélisation de μLJ	166
6.3.1	Catégories μ -fermées	166
6.3.2	Lemmes de substitution	167
6.4	Interprétation	168
6.4.1	Interprétation d'une preuve	168
6.4.2	Correction de l'interprétation	171
7	Foncteurs ouverts	173
7.1	Composition verticale	175
7.1.1	Arènes ouvertes	175
7.1.2	Substitution d'arènes	176
7.1.3	Parties fonctorielles	178
7.1.4	Stratégies fonctorielles	181
7.1.5	Restriction verticale	182
7.1.6	Composition verticale	187
7.1.7	Décomposition des stratégies fonctorielles	191
7.2	Foncteurs ouverts et stratégies naturelles	193
7.2.1	Loi d'échange	194
7.2.2	Restriction et composition	196
7.2.3	Foncteurs usuels	199
7.3	Théorème de factorisation	199
7.3.1	Trivialité pour l'identité	200
7.3.2	Factorisation	204
7.4	Transformations dinaturelles ouvertes	209
7.4.1	Échange dinaturel	209
7.4.2	Stabilité par composition	210
7.4.3	Sur la factorisation.	212
7.5	Types Forts	212
7.5.1	Foncteurs ouverts et forces	212
7.5.2	Structure de types forts	214
7.5.3	Foncteur ouverts et types rékursifs	216
8	Modèle des points fixes	219
8.1	Jeux à gain	220
8.1.1	Décomposition en fils	220
8.1.2	Jeux à gain	221
8.1.3	La catégorie Gain	223
8.1.4	Relations avec la Partie I	225
8.2	Points fixes	227
8.2.1	Foncteurs ouverts à gain	227
8.2.2	Types forts pour le gain	228
8.2.3	Construction de boucle	229
8.2.4	Une catégorie μ -fermée	233
8.3	Sur la complétude	236
	Conclusions et perspectives	239

Introduction

Problématique

Les travaux présentés dans cette thèse sont à l'interface entre la *sémantique dénotationnelle*, qui étudie les programmes informatiques par des représentations mathématiques invariantes par réduction et la *logique mathématique* qui étudie par divers outils les raisonnements et principes logiques et leur représentations syntaxiques. Ces deux domaines d'étude sont mis en relation par la désormais célèbre *correspondance de Curry-Howard*, c'est donc par elle qu'on va commencer cette introduction.

Correspondance de Curry-Howard. À l'origine, la *correspondance de Curry-Howard* vient d'un travail de Howard [How80] qui, s'inspirant de travaux précédents de Curry sur la logique minimale à la Hilbert et les *combinateurs* S et K , releva que les arbres de preuve en déduction naturelle intuitionniste pouvaient être vus comme des dérivations de typage pour des termes du λ -calcul. En généralisant à partir de cette situation, il est devenu naturel de qualifier de *correspondance de Curry-Howard* toute situation où un système de preuves coïncide avec une notion de typage pour un langage donné. Bien que fort intéressante, cette vision où preuves et programmes persistent à vivre dans des univers séparés trouve rapidement ses limites. Par exemple, que peut-on dire d'un langage typé «à la Church», où les termes portent avec eux les annotations de type ?

Sémantique dénotationnelle. La *sémantique dénotationnelle* tente de donner du sens aux programmes et aux preuves en leur associant des *invariants*. Ainsi, elle associe à chaque preuve ou à chaque programme une représentation abstraite, par exemple une fonction continue entre domaines [Sco82] ou une stratégie sur un jeu bien choisi [HO00]. Une des particularités de la sémantique dénotationnelle est qu'elle interprète uniformément preuves et programmes. Qui plus est, elle les plonge dans le même univers. Ainsi, une catégorie cartésienne fermée peut être utilisée à la fois pour représenter la logique intuitionniste avec conjonction et les termes simplement typés du λ -calcul. Les programmes et les preuves sont représentés par les mêmes objets. Ainsi étant donné un modèle, on peut se poser la question : «comment ce modèle distingue-t-il preuves et programmes ?»

Preuves et jeux. Posons-nous cette question pour les jeux. Quelles propriétés partagent les stratégies qui viennent de l'interprétation de preuves ? Notons qu'un système de preuve plausible doit vérifier une propriété de cohérence, qui est en général obtenue via la *propriété de la sous-formule*, elle-même conséquence de l'*élimination des coupures* [GTL89]. Ainsi, la propriété fondamentale des stratégies provenant de preuves devrait être une propriété des in-

interprétations de preuves sans coupure. Autrement dit, ces stratégies sont obtenues en utilisant uniquement les constructeurs de la structure cartésienne fermée mais sans composition. Il est immédiat que les stratégies obtenues de cette façon sont *totales*.

Dialogue infini. Il est donc naturel d'interpréter les preuves par des stratégies totales. Mais pour qu'une telle classe soit adéquate pour représenter les preuves, il ne suffit pas qu'elle soit une propriété commune de toutes les interprétations de preuves ; il faut également qu'elle soit invariante par toutes les opérations usuelles, y compris la composition. Il faut donc s'assurer que la composition de deux stratégies totales est toujours totale. Malheureusement, cela est faux en général : il peut se produire un problème de *dialogue infini*, similaire au phénomène de *livelock* qu'on peut observer dans les modèles de la concurrence. L'idée est que deux processus ou stratégies peuvent s'engager dans un dialogue infini dans leur zone de communication privée, dialogue qui n'est jamais vu depuis l'extérieur. On présentera dans cette thèse (Figure 4.3) un exemple très concret d'interaction infinie entre deux stratégies totales, produisant une stratégie partielle. Le problème de la modélisation des systèmes logiques en sémantique de jeux est donc en étroite relation avec celui de trouver des classes de stratégies pour lesquelles le caractère total est stable par composition. Un bref examen de la littérature propose diverses solutions à ce problème.

Complétude. La plus répandue s'appuie sur le fait que les jeux parviennent fréquemment à caractériser les preuves de manière *pleinement complète*. Autrement dit, chaque stratégie correspond à une preuve, parfois de façon unique. Le raisonnement fonctionne donc comme suit. On considère deux stratégies σ et τ , qu'on fait communiquer sur un canal B . Notons $\sigma; \tau$ la stratégie obtenue par composition¹, on voudrait pouvoir assurer qu'elle est totale. Par la propriété de complétude, on peut trouver des preuves π_1 et π_2 antécédentes de σ et τ pour l'interprétation, c'est à dire telles que $\sigma = \llbracket \pi_1 \rrbracket$ et $\tau = \llbracket \pi_2 \rrbracket$. Il suffit alors de noter que $\sigma; \tau$ aura un antécédent donné par la notion de composition équivalente dans la syntaxe, que l'on notera ici $\pi_1; \pi_2$. Mais notre système de preuve normalise, ainsi $\pi_1; \pi_2$ est équivalente à une preuve sans coupure qui s'interprète par une stratégie totale. Ainsi, la validité du modèle (duquel on attendrait qu'il justifie la syntaxe, qu'il lui donne un sens) dépend du résultat de normalisation dans la syntaxe. Si cette approche convient souvent d'un point de vue technique, on peut déplorer son pauvre contenu épistémologique.

Gain. Une autre solution, plus satisfaisante, est donnée par la méthode du *gain*, très similaire aux méthodes du même nom utilisées dans les jeux de vérification pour contrôler les parties infinies. À chaque partie infinie, on associe un *gagnant* et un *perdant*. Les stratégies sont désormais supposées gagnantes, ainsi les seuls dialogues infinis auxquels elles prendront part seront causés par une défaillance de leur adversaire. Le problème est donc de savoir désigner un *responsable* en cas de divergence. Il n'y a en général pas de façon canonique de définir le gain, ainsi plusieurs fonctions de gain sur un même jeu pourront donner lieu à des représentations de plusieurs formules ou types de données : il n'y a pas toujours, étant donné un jeu A , de façon canonique de désigner un responsable pour les divergences sur A . C'est donc la formule représentée par A qui va diriger le choix de ce responsable : la fonction de gain sera en général définie par induction sur la formule. En cela elle est à rapprocher de la notion de réalisabilité, qui s'appuie sur la formule interprétée pour établir les responsabilités en cas de non terminaison

¹Pour le moment, ces notations sont à prendre informellement puisqu'on n'a encore rien défini.

d'une réduction. Chaque atome (*i.e.* règle de déduction de la logique) étant «responsable», on déduit alors la normalisation de l'ensemble (*i.e.* de toutes les preuves engendrées par ces règles).

Normalisation intrinsèque. Une autre solution, qui semble de prime abord diamétralement opposée, consiste à oublier dans un premier temps les formules et à rechercher à la place des propriétés intrinsèques sur les stratégies permettant d'assurer leur stabilité par composition. On donne ainsi un critère de *finitude* sur les stratégies (parfois également appelé gain), et on montre que deux stratégies vérifiant cette propriété de finitude ne peuvent pas avoir d'interaction infinie. Cette approche est moins fréquente, l'auteur ne connaît que deux exemples de son utilisation dans un contexte associé aux jeux : on la voit à l'œuvre dans les travaux de Coquand [Coq95] puis Curien [Cur98]. Si la construction d'une sémantique des preuves a pour objet leur explication par les objets plus «élémentaires» que sont les preuves, alors cette approche est la plus satisfaisante : en effet, le modèle ne s'appuie pas sur la structure ou les propriétés des termes ou des types pour assurer cette stabilité de la totalité. En un sens (que nous rendrons précis dans cette thèse), c'est la normalisation dans la syntaxe qui découle de propriétés des stratégies de l'interprétation, et non l'inverse.

Première partie

La première partie de cette thèse s'inscrit directement dans le prolongement de cette problématique : le but est d'obtenir et d'étudier des propriétés intrinsèques sur les stratégies permettant d'assurer la stabilité de la totalité par composition.

Chapitre 3. On y explore les connexions entre ce problème de stabilité de la totalité par composition, et les résultats de normalisation usuels en λ -calcul. On introduit le *λ -calcul unaire*, un λ -calcul simplement typé et non déterministe extrêmement simple pour lequel on prouve un résultat de normalisation par des méthodes usuelles. On définit pour ce calcul une notion de réduction linéaire de tête, qu'on relie à l'exécution de différentes machines abstraites (notamment la machine de Krivine et la machine à Pointeurs). Dans tout ce chapitre, on introduira tout doucement la relation qu'entretient l'exécution de ce calcul avec l'interaction entre stratégies.

Chapitre 4. Ce chapitre se concentre essentiellement sur la notion de *structure de pointeurs* : il s'agit de parties légales sur lesquelles on a oublié toute identification sur les coups. C'est à ces structures axiomatiques que l'on veut réduire le problème de stabilité de la totalité par composition. Après avoir étudié quelques-unes de leurs propriétés, on énonce les *théorèmes de finitude*, principaux résultats de cette partie. On en donne trois preuves différentes (même s'il est possible d'établir des liens entre les techniques utilisées). La première est la seule à utiliser explicitement la structure développée au Chapitre 3 (pour les autres, il s'agira surtout d'une inspiration) : on utilise une variante de la PAM pour ramener le problème de finitude à la normalisation de la réduction linéaire de tête pour le λ -calcul unaire. La seconde est une internalisation de la preuve précédente dans le vocabulaire des structures de pointeurs ; en particulier cela permet d'extraire une réduction élémentaire sur les arbres d'entiers dont la combinatoire est sous-jacente à la réduction linéaire de tête des λ -calculs typés. La dernière preuve est une adaptation de la preuve de Coquand [Coq95] au vocabulaire des structures

de pointeurs. Finalement, on tire de ces théorèmes de finitude trois classes de stratégies pour lesquelles la totalité est préservées par composition : les stratégies *finies*, *bornées* et *nœthériennes*.

Deuxième partie

L'approche décrite ci-dessus peut permettre de donner des preuves de stabilité de la totalité par composition pour les modèles de jeux de nombreuses logiques, mais atteint ses limites dès qu'on veut représenter des types de données non bornés : par exemple, on verra que l'identité sur les entiers paresseux ne vérifie aucune des conditions définies plus haut². La deuxième partie se veut apporter une pierre à l'étude des cas non couverts par la première partie. Cela se fera par l'intermédiaire de l'étude d'une logique μLJ adaptée de [BM07], comportant des constructeurs pour les structures de données inductives et coinductives. Puisque l'approche de la première partie ne suffit plus, on fera appel au *gain*. On tire inspiration des conditions de gains des *jeux de parité* : ces conditions permettent de représenter de façon fiable les structures de données définies par des μ -termes [San02a], et on montre ici qu'on peut les adapter au cadre de la sémantique des jeux pour intégrer le connecteur flèche.

Chapitre 5. On commence par introduire la logique μLJ : il s'agit d'une extension directe des séquents intuitionnistes LJ de Gentzen par des règles μ et ν pour les plus petits et plus grands points fixes. Cette logique peut être vue comme une base pour un système de preuve intégrant des raisonnements inductifs et coinductifs, ou comme un langage de programmation normalisant, travaillant sur les plus petits et plus grands points fixes. Ce dernier point de vue nous incitera à introduire des règles de déduction sur les points fixes modulo un contexte Γ , pour pouvoir modéliser l'itération d'une fonction comportant des variables libres. Finalement, on lui associe une notion de réduction, qu'on justifie en montrant qu'elle donne lieu à une simulation du système T .

Chapitre 6. On établit ici la structure catégorique nécessaire pour donner un modèle correct de μLJ . Ces travaux s'appuient sur la représentation habituelle des plus petits et plus grands points fixes par des algèbres initiales et coalgèbres terminales aux foncteurs définis par les formules. Notons la complication due aux règles de point fixe modulo un contexte Γ mentionnées ci-dessus, qui nous inciteront à équiper les relèvements de foncteurs définis par la syntaxe à l'aide de *forces*. On définit ainsi les *catégories μ -fermées*, qui sont des modèles corrects pour μLJ .

Chapitre 7. L'inspiration des jeux de parité nous incite à représenter les points fixes de façon finie par des *boucles* dans les arènes. Ce choix de représentation motive le développement de la notion de *foncteur ouvert*, donnant une action fonctorielle à toute arène munie de variables libres. On montrera que ces foncteurs ouverts héritent naturellement de forces, faisant d'eux de bons candidats pour supporter l'interprétation de μLJ .

²Par contre, tout se passe bien pour les entiers *plats*. En pratique, tout se qui compte est la profondeur de l'arène, sa largeur importe peu.

Chapitre 8. On introduit finalement la notion de *jeu à gain* et de *stratégies gagnantes*. Une légère digression nous permet d'établir des ponts entre cette notion de gain et les critères de normalisation intrinsèques développés dans la première partie, puis on poursuit la construction du modèle des points fixes. On introduit une *construction de boucle* sur les arènes, autorisant la représentation finie de structures de données infinies. Puis on équipe ces boucles de conditions de gain inspirées des jeux de parité, donnant ainsi des algèbres initiales et coalgèbres terminales aux foncteurs ouverts. On montre qu'on définit bien ainsi une catégorie μ -fermée et donc un modèle de μLJ . On se pose alors la question de la complétude de ce modèle, à laquelle on donne quelques éléments de réponse.

Préliminaires

Chapitre 1

Théorie des catégories

Sommaire

1.1	Catégories	20
1.1.1	Définitions basiques	20
1.1.2	Foncteurs, transformations naturelles	22
1.2	Structure catégorique du calcul	23
1.2.1	Structure bicartésienne	23
1.2.2	Catégories cartésiennes fermées	25
1.2.3	Coproduits faibles fonctoriels	26
1.3	Préliminaires spécifiques	27
1.3.1	Algèbres initiales et coalgèbres finales	27
1.3.2	Catégories de co-Kleisli et relèvements	29

Lorsqu'en mathématiques on définit une nouvelle classe d'objets, le réflexe souvent naturel est de définir en même temps les *morphismes* entre ces objets, c'est-à-dire les transformations qui préservent la structure de ces objets : souvent, ces morphismes produisent de puissants outils pour raisonner sur les objets de départ. Par exemple, on pourra faire apparaître d'importantes propriétés d'un espace topologique en examinant par quelles applications continues il se plonge dans d'autres espaces. Ou encore, on pourra comparer les dimensions de deux espaces vectoriels en observant les applications linéaires entre l'un et l'autre.

La *théorie des catégories* peut être perçue comme la systématisation de cette approche. Au lieu de considérer d'abord des objets qui génèrent ensuite des morphismes, on considère donnée une classe d'objets et de morphismes, et les seules propriétés qu'on pourra alors considérer sur ces objets seront dérivées à partir des morphismes. Autrement dit, on oublie tout de la structure des objets pour se concentrer sur la question : «Quelles sont les interactions entre ces objets, comment se transforment-ils l'un en l'autre?». Cette approche s'est révélée très productive dans de nombreux champs des mathématiques, notamment en informatique.

Pour que cette thèse soit la plus complète possible, on va introduire certains concepts de théorie des catégories qui nous seront utiles le long de cet ouvrage. Pour les lecteurs plus intéressés, un livre de référence est [\[ML71\]](#).

1.1 Catégories

1.1.1 Définitions basiques

Donnons ici les définitions de base des catégories ainsi que les concepts s'appuyant uniquement sur cette structure basique.

Définition 1.1.1. Une catégorie \mathcal{C} est la donnée de :

- Une classe d'objets \mathcal{C}_0 ;
- Pour tous $A, B \in \mathcal{C}_0$, un ensemble de flèches ou de morphismes $\mathcal{C}(A, B)$. Pour $f \in \mathcal{C}(A, B)$, on notera $f : A \rightarrow B$;
- Pour chaque objet $A \in \mathcal{C}_0$, une identité $id_A : A \rightarrow A$;
- Une opération de composition qui à $f : A \rightarrow B$ et $g : B \rightarrow C$ associe $(f; g) : A \rightarrow C$;

Toutes ces données doivent vérifier les axiomes suivants :

- *Associativité* : pour tous objets A, B, C, D , pour tous $f : A \rightarrow B$, $g : B \rightarrow C$ et $h : C \rightarrow D$, on a

$$(f; g); h = f; (g; h)$$

- *Neutralité des identités* : pour tout morphisme $f : A \rightarrow B$, on a les équations :

$$\begin{aligned} id_A; f &= f \\ f; id_B &= f \end{aligned}$$

En cas d'ambiguïté, on annote les opérations de composition et les identités par le nom de la catégorie correspondante, c'est-à-dire qu'on écrit $id_A^{\mathcal{C}}$ ou $f;^{\mathcal{C}}g$.

Remarque. On appelle *petites* les catégories où la classe des objets est en fait un ensemble. Dans cette thèse, la grande majorité des catégories utilisées seront petites : l'ensemble des objets sera même la plupart du temps dénombrable. Occasionnellement, on aura tout de même besoin de parler de catégories dont les objets ne forment pas un ensemble, par exemple la catégorie des petites catégories (à venir). Notons que les catégories définies ci-dessus sont toutes *localement petites*, c'est à dire que $\mathcal{C}(A, B)$ est un ensemble et jamais une classe propre.

Exemples. Donnons d'abord quelques cas particuliers de catégories :

- Tout monoïde \mathcal{M} peut être vu comme une catégorie à un objet \star : les morphismes $a : \star \rightarrow \star$ sont les éléments du monoïde, la composition est la multiplication. Les catégories sont donc une généralisation des monoïdes : une catégorie est un «monoïde typé».
- Tout ensemble partiellement ordonné (E, \leq) peut être vu comme une catégorie. Les objets sont les éléments de E , et l'ensemble des morphismes de x à y est le singleton $\{\rightarrow\}$ si $x \leq y$, l'ensemble vide sinon. Les catégories sont donc une généralisation des ordres partiels.

Les catégories «concrètes» abondent dans la littérature : il suffit de sélectionner une structure mathématique et une classe de morphismes la préservant. En voici quelques exemples :

- la catégorie **Ens** des ensembles et fonctions ;
- la catégorie des groupes et morphismes de groupes ;
- la catégorie des graphes et morphismes de graphes ;
- la catégorie des espaces topologiques et applications continues ;

- la catégorie des espaces vectoriels et applications linéaire ;
- ...

Définition 1.1.2 (Sous-catégorie). *Soient \mathcal{C} et \mathcal{D} deux catégories. On dit que \mathcal{C} est une sous-catégorie de \mathcal{D} si :*

- $\mathcal{C}_0 \subseteq \mathcal{D}_0$;
- Pour tous $A, B \in \mathcal{C}_0$, $\mathcal{C}(A, B) \subseteq \mathcal{D}(A, B)$;
- Les deux catégories ont les mêmes identités : pour tout $A \in \mathcal{C}_0$, $id_A^{\mathcal{C}} = id_A^{\mathcal{D}}$;
- Les deux catégories ont la même composition : pour tous $f : A \rightarrow B$, $g : B \rightarrow C$ dans \mathcal{C} , $f;^{\mathcal{C}}g = f;^{\mathcal{D}}g$.

Définition 1.1.3 (Mono, épi). *Soit \mathcal{C} une catégorie. Un morphisme $f : A \rightarrow B$ est un monomorphisme s'il est simplifiable à droite, c'est à dire si pour tout objet C et morphismes $g_1, g_2 : C \rightarrow A$, si $g_1;f = g_2;f$ alors $g_1 = g_2$. De même, f est un épimorphisme s'il est simplifiable à gauche.*

Les notions de monomorphisme et d'épimorphisme sont des abstractions des notions ensemblistes de fonction injectives ou surjective. Et en effet, on peut vérifier que les monomorphismes de la catégorie des ensembles sont bien les fonctions injectives et que les épimorphismes en sont bien les fonctions surjectives. Ces définitions capturent bien de façon abstraite les propriétés assurées par la surjectivité ou l'injectivité. Par contre, il n'est pas satisfaisant de définir un *isomorphisme* par un morphisme à la fois mono et epi : on n'aurait pas toujours l'existence de l'inverse.

Définition 1.1.4. *Un morphisme $f : A \rightarrow B$ est un isomorphisme s'il existe $f^{-1} : B \rightarrow A$ tel que $f;f^{-1} = id_A$ et $f^{-1};f = id_B$.*

Diagrammes commutatifs. En théorie des catégories, on représente souvent des relations entre morphismes par des *diagrammes commutatifs*. Un diagramme est donné par une représentation d'un graphe fini orienté, et sa commutativité équivaut au fait que tous ses chemins parallèles (*i.e.* partant et arrivant au même point) sont égaux. Par exemple, le diagramme commutatif suivant

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ \downarrow g & & \downarrow h \\ C & \xrightarrow{k} & D \end{array}$$

exprime l'égalité entre morphismes suivante :

$$f;h = g;k$$

Constructions. Donnons ici quelques constructions usuelles sur les catégories.

Définition 1.1.5 (Catégorie opposée). *Soit \mathcal{C} une catégorie. Sa catégorie opposée, notée \mathcal{C}^{op} , est définie par les données suivantes :*

- Ses objets sont les objets de \mathcal{C} ;
- L'ensemble des morphismes $\mathcal{C}^{op}(A, B)$ est défini par $\mathcal{C}(B, A)$.

La définition des identités et de la composition en découle immédiatement.

Définition 1.1.6 (Catégorie produit). Soient \mathcal{C} et \mathcal{D} deux catégories. La catégorie produit $\mathcal{C} \times \mathcal{D}$ est définie par :

- Ses objets sont les couples (A, B) où $A \in \mathcal{C}_0$ et $B \in \mathcal{D}_0$;
- Un morphisme de (A, B) vers (C, D) est un couple (f, g) où $f : A \rightarrow C$ (dans \mathcal{C}) et $g : B \rightarrow D$ (dans \mathcal{D}).

La définition des identités et de la composition en découle immédiatement.

1.1.2 Foncteurs, transformations naturelles

Foncteurs. Puisque le slogan principal des catégories est de pousser au second plan les objets et de mettre au premier plan les morphismes entre objets, appliquons ce slogan aux catégories elles-mêmes et donnons la notion appropriée de morphismes entre catégories : il s'agit des *foncteurs*.

Définition 1.1.7 (Foncteur). Soient \mathcal{C} et \mathcal{D} deux catégories. Par abus de notation, on va noter de la même façon les identités et la composition sur \mathcal{C} et sur \mathcal{D} . Un foncteur $F : \mathcal{C} \rightarrow \mathcal{D}$ est la donnée des éléments suivants :

- Une fonction $F_0 : \mathcal{C}_0 \rightarrow \mathcal{D}_0$;
- Pour tout objets $A, B \in \mathcal{C}_0$, une fonction $F_{A,B} : \mathcal{C}(A, B) \rightarrow \mathcal{D}(F_0(A), F_0(B))$.

En général, on se contente de noter F toutes les composantes de F . Ces opérations doivent alors préserver la structure des catégories, c'est-à-dire :

- Pour tout objet A , $F(id_A^{\mathcal{C}}) = id_{F(A)}^{\mathcal{D}}$;
- Pour tous morphismes $f : A \rightarrow B$ et $g : B \rightarrow C$, $F(f;^{\mathcal{C}}g) = F(f);^{\mathcal{D}}F(g)$.

Exemple. Si \mathbb{K} est un corps, considérons la catégorie **Vect** des \mathbb{K} -espaces vectoriels et applications linéaires, et prenons $F_0 : \mathbf{Ens}_0 \rightarrow \mathbf{Vect}_0$ qui associe à chaque ensemble E le \mathbb{K} -espace vectoriel $F_0(E)$ librement engendré par E . La fonction F_0 s'étend alors en un foncteur $F : \mathbf{Ens} \rightarrow \mathbf{Vect}$ qui associe à chaque fonction $f : A \rightarrow B$ l'unique application linéaire $F_0(A) \rightarrow F_0(B)$ engendrée par f .

Définition 1.1.8 (Cat). On définit la catégorie **Cat** des petites catégories, dont les objets sont les petites catégories et les morphismes sont les foncteurs.

Remarque. En sémantique des langages de programmation, les foncteurs modélisent fréquemment des types à variables libres. On rappellera dans la Partie II comment tout type à variables libres définit un foncteur.

Transformations naturelles. On peut appliquer le même slogan sur les foncteurs eux-mêmes et tâcher de trouver la notion appropriée de morphismes entre foncteurs : il s'agit des *transformations naturelles* :

Définition 1.1.9 (Transformation naturelle). Soient $F, G : \mathcal{C} \rightarrow \mathcal{D}$ deux foncteurs. Une transformation naturelle η de F dans G , notée $\eta : F \overset{\bullet}{\rightarrow} G$, est une famille de morphismes $(\eta_A)_{A \in \mathcal{C}_0} : F(A) \rightarrow G(A)$ telle pour tout $f : A \rightarrow B$ dans \mathcal{C} , le diagramme suivant commute :

$$\begin{array}{ccc} F(A) & \xrightarrow{\eta_A} & G(A) \\ F(f) \downarrow & & \downarrow G(f) \\ F(B) & \xrightarrow{\eta_B} & G(B) \end{array}$$

Remarque. Si on voit les foncteurs comme une abstraction des types à variables libres, on peut voir une transformation naturelle comme une notion de *terme polymorphe* en ces variables [BFSS90].

Adjonctions. Un des principaux apports des définitions précédentes est de permettre d'exprimer la notion d'*adjonction* : c'est la notion catégorique qui généralise celle de *correspondance de Galois* dans les ordres partiels.

Définition 1.1.10. Soient $F : \mathcal{C} \rightarrow \mathcal{D}$ et $G : \mathcal{D} \rightarrow \mathcal{C}$ deux foncteurs. On dit que F est adjoint à gauche de G (resp. G est adjoint à droite de F), noté $F \dashv G$, si on a une famille de bijections :

$$\mathcal{D}(F(A), B) \cong \mathcal{C}(A, G(B))$$

naturelle en A et en B , c'est à dire qu'il s'agit d'un isomorphisme naturel $\phi : \mathcal{D}(F(-), -) \xrightarrow{\bullet} \mathcal{C}(-, G(-))$.

1.2 Structure catégorique du calcul

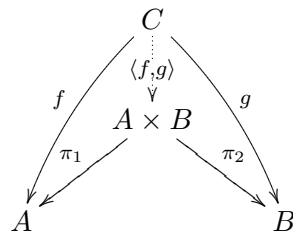
Dans cette section, nous introduisons la structure nécessaire pour interpréter une logique intuitionniste (ou un langage de programmation). Pour chacune de ces constructions on donnera à chaque fois la définition traditionnelle (axiomatique, par propriété universelle), puis on en donnera une approche équationnelle [LS88].

1.2.1 Structure bicartésienne

La première étape est d'introduire la structure *bicartésienne*, permettant d'interpréter le produit et la somme.

Structure cartésienne. Introduisons les produits, et l'objet terminal.

Définition 1.2.1 (Produits). Soit \mathcal{C} une catégorie et soient $A, B \in \mathcal{C}_0$. Un produit de A et B est un objet $A \times B \in \mathcal{C}_0$ muni de deux projections $\pi_1 : A \times B \rightarrow A$ et $\pi_2 : A \times B \rightarrow B$ et vérifiant la propriété universelle suivante : Pour tout $f : C \rightarrow A$, pour tout $g : C \rightarrow B$, il existe un unique morphisme $\langle f, g \rangle : C \rightarrow A \times B$ faisant commuter le diagramme suivant :



Donnons une présentation équationnelle équivalente, à la saveur plus «syntaxique» : supposons que pour deux objets A et B , on ait un objet $A \times B$ muni de projections $\pi_1 : A \times B \rightarrow A$

et $\pi_2 : A \times B \rightarrow B$, et d'une fonction associant à tous morphismes $f : C \rightarrow A$ et $g : C \rightarrow B$ un morphisme $\langle f, g \rangle : C \rightarrow A \times B$, le tout vérifiant les équations suivantes :

$$\begin{aligned}\langle f, g \rangle; \pi_1 &= f \\ \langle f, g \rangle; \pi_2 &= g \\ \langle \pi_1; \pi_2 \rangle &= id \\ h; \langle f, g \rangle &= \langle h; f, h; g \rangle\end{aligned}$$

Alors il est immédiat de vérifier que ces données donnent à $A \times B$ une structure de produit.

Définition 1.2.2 (Objet terminal). *Si \mathcal{C} est une catégorie, un objet $1 \in \mathcal{C}_0$ est terminal si pour tout $A \in \mathcal{C}_0$ il existe un unique morphisme $!_A : A \rightarrow 1$. Notons que si \mathcal{C} a un objet terminal 1 , alors celui-ci est unique à isomorphisme près.*

Définition 1.2.3 (Catégorie cartésienne). *Une catégorie \mathcal{C} est cartésienne si elle admet un objet terminal 1 et que tous objets A et B ont un produit $A \times B$ dans \mathcal{C} .*

Remarque. Une remarque importante est que dans toute catégorie cartésienne \mathcal{C} , \times devient un foncteur $\mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$. Son image sur les objets est le produit, et $f \times g = \langle \pi_1; f, \pi_2; g \rangle$: une vérification immédiate montre que cela définit bien un foncteur.

Structure bicartésienne. Une catégorie bicartésienne est une catégorie cartésienne équipée de sommes :

Définition 1.2.4 (Coproducts). *Soit \mathcal{C} une catégorie et soient $A, B \in \mathcal{C}_0$. Un coproduit de A et B est un objet $A + B \in \mathcal{C}_0$ muni de deux injections $in_1 : A \rightarrow A + B$ et $in_2 : B \rightarrow A + B$ et vérifiant la propriété universelle suivante : Pour tous $f : A \rightarrow C$ et $g : B \rightarrow C$, il existe un unique morphisme $[f|g] : A + B \rightarrow C$ faisant commuter le diagramme suivant :*

$$\begin{array}{ccc} A & & B \\ & \searrow^{in_1} & \swarrow_{in_2} \\ & A + B & \\ & \searrow^f & \swarrow_g \\ & & C \end{array}$$

$\downarrow [f|g]$

Si on relâche la condition d'unicité de $[f|g]$, on obtient la notion de coproduit faible. Dans cette thèse, on utilisera occasionnellement la terminologie de coproduit strict pour insister sur le caractère non faible d'un coproduit.

Donnons cette fois encore une présentation équationnelle du coproduit. Supposons qu'on ait deux objets A, B , un objet $A + B$, deux injections in_1 et in_2 et une fonction qui à tous $f : A \rightarrow C$ et $g : B \rightarrow C$ associe un morphisme $[f|g] : A + B \rightarrow C$, le tout vérifiant les équations suivantes :

$$\begin{aligned}in_1; [f|g] &= f \\ in_2; [f|g] &= g \\ [in_1|in_2] &= id_{A+B} \\ [f|g]; h &= [f; h|g; h]\end{aligned}$$

Alors ces données donnent à $A + B$ une structure de coproduit.

Définition 1.2.5 (Objet initial). *Si \mathcal{C} est une catégorie, un objet $0 \in \mathcal{C}_0$ est initial si pour tout $A \in \mathcal{C}_0$ il existe un unique morphisme $i_A : 0 \rightarrow A$. On dit que 0 est faiblement initial si on oublie la condition d'unicité, c'est à dire qu'il peut y avoir plusieurs morphismes de 0 vers A .*

Définition 1.2.6 (Catégorie bicartésienne). *Une catégorie \mathcal{C} est bicartésienne si elle est cartésienne, que tous objets A, B admettent un coproduit $A + B$ et qu'elle a un objet initial.*

Remarque. Comme pour le produit, $+$ est alors lui aussi équipé d'une structure de foncteur $+: \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$. Son image sur les objets est leur coproduit, et $f + g = [in_1; g|in_2; g]$.

1.2.2 Catégories cartésiennes fermées

Pour interpréter un calcul de fonctions, il est nécessaire d'avoir la structure nécessaire pour *internaliser* les espaces de fonctions, c'est-à-dire avoir un objet $A \Rightarrow B$ représentant, en un certain sens, les morphismes $A \rightarrow B$.

Définition 1.2.7 (CCC). *Soit \mathcal{C} une catégorie cartésienne. \mathcal{C} est cartésienne fermée si le foncteur $A \times - : \mathcal{C} \rightarrow \mathcal{C}$ a un adjoint à droite $A \Rightarrow - : \mathcal{C} \rightarrow \mathcal{C}$.*

Comme précédemment, on va en donner une présentation équationnelle. La condition d'adjonction s'exprime par le fait qu'on a une famille de bijections, naturelle en A et en B :

$$A \times B \rightarrow \mathcal{C} \cong B \rightarrow (A \Rightarrow \mathcal{C})$$

Notons Λ cet isomorphisme, et définissons $ev_{A,B} : A \times (A \Rightarrow B) \rightarrow B$ par $ev = \Lambda^{-1}(id)$. Le fait que \mathcal{C} soit cartésienne fermée est alors équivalent à la satisfaction des deux équations suivantes :

$$\langle \pi_1, \pi_2; \Lambda(f) \rangle; ev = f \tag{1.1}$$

$$\Lambda(\langle \pi_1, \pi_2; g \rangle; ev) = g \tag{1.2}$$

Lemme 1.2.1. *Si \mathcal{C} est cartésienne fermée, les équations ci-dessous sont également vérifiées.*

$$\langle g, \Lambda(f) \rangle; ev = \langle g, id \rangle; f \tag{1.3}$$

$$g; \Lambda(f) = \Lambda(\langle \pi_1, \pi_2; g \rangle; f) \tag{1.4}$$

$$\Lambda(ev) = id \tag{1.5}$$

Interaction avec les sommes. La structure cartésienne fermée n'est pas indépendante des sommes, et les interactions entre ces différentes composantes provoquent des phénomènes intéressants.

Proposition 1.2.2. *Si \mathcal{C} est bicartésienne fermée, alors \times distribue sur $+$. Plus formellement, on a un isomorphisme naturel :*

$$\gamma_{\Gamma, A, B} : \Gamma \times (A + B) \rightarrow \Gamma \times A + \Gamma \times B$$

Idée de preuve. Avec la syntaxe donnée ci-dessus, le morphisme $\gamma_{\Gamma,A,B}$ est donné par :

$$\gamma_{\Gamma,A,B} = \Lambda^{-1}([\Lambda(in_1)|\Lambda(in_2)])$$

Son inverse $\gamma_{\Gamma,A,B}^{-1}$ est définissable sans recours à la structure fermée, par :

$$\gamma_{\Gamma,A,B}^{-1} = \langle [\pi_1|\pi_1], [\pi_2; in_1|\pi_2; in_2] \rangle$$

□

Cette *loi de distributivité* est importante, car c'est elle qui permet de modéliser le raisonnement conditionnel. En particulier, elle est cruciale pour raisonner par cas sur les valeurs d'un terme contenant des variables libres (*i.e.* contenues dans Γ). C'est donc une bonne propriété. Malheureusement, les catégories bicartésiennes fermées ont également la propriété déconcertante suivante, les rendant peu appropriées pour servir de base à des modèles de jeux (outre le fait qu'il est difficile de construire une catégorie de jeux équipée à la fois de produits et de coproduits, voir la discussion dans la Section 2.2.6).

Proposition 1.2.3. *Dans une catégorie bicartésienne fermée, il y a au plus un morphisme $A \rightarrow 0$.*

Démonstration. Preuve dans [LS88] (Proposition 8.3). □

Cette proposition rend les catégories bicartésiennes fermées inappropriées pour servir de base à des modèles de jeux, parce qu'elles écrasent trop de comportement intensionnel. En effet, et en anticipant quelque peu sur le modèle de jeux, il est facile de trouver une arène A pour laquelle il existe plusieurs stratégies totales $\sigma : A \Rightarrow 0$: il suffit par exemple de prendre $A = 0 \times 0$. Opposant dispose de deux «preuves» de 0, et Joueur peut choisir d'interroger l'une ou l'autre.

1.2.3 Coproduits faibles fonctoriels

Il nous faut donc une notion de somme plus souple que les coproduits, qui permette tout autant d'interpréter les sommes des langages de programmation mais qui satisfasse moins d'équations. Bien sûr, il sera toujours crucial que cette définition affaiblie de somme donne lieu à la construction d'un foncteur $+$: $\mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$. Notons que nous n'avons pas trouvé de référence bibliographique pour la variante précise des sommes que nous allons présenter ici, mais de nombreux affaiblissements des coproduits ont été considérés dans la littérature : citons par exemple les *sommes strictes* [McC96].

Définition 1.2.8. *Une catégorie \mathcal{C} a les coproduits faibles fonctoriels si elle est équipée des données suivantes :*

- Pour tous objets A et B , un objet $A + B$ et deux morphismes $in_1 : A \rightarrow A + B$ et $in_2 : B \rightarrow A + B$;
- Pour tous morphismes $f : A \rightarrow C$ et $g : B \rightarrow C$, un morphisme $[f|g] : A + B \rightarrow C$;
- Un objet faiblement initial 0 et pour tout objet A , un morphisme $i_A : 0 \rightarrow A$.

Ces données doivent vérifier les équations suivantes :

$$\begin{aligned} in_1; [f|g] &= f \\ in_2; [f|g] &= g \\ [in_1|in_2] &= id \\ [f; in_1|g; in_2]; [f'|g'] &= [f; f'|g; g'] \end{aligned}$$

Ces équations ne suffisent pas à assurer qu'on a un coproduit, c'est donc en général un *coproduit faible*. Le long de cette thèse on n'utilisera que cette structure de coproduits faibles, mais les résultats seront évidemment toujours valides en présence d'un véritable coproduit.

Proposition 1.2.4. *Si \mathcal{C} a les coproduits faibles fonctoriels, alors on a bien un foncteur $+$: $\mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$.*

Démonstration. Calculs directs. □

Distributivité. Comme dans le cas des coproduits, la structure fermée permet de construire une distributivité du produit sur la somme. On peut vérifier que la même définition que précédemment donne toujours une transformation naturelle :

$$\gamma_{\Gamma, A, B} : \Gamma \times (A + B) \rightarrow \Gamma \times A + \Gamma \times B$$

1.3 Préliminaires spécifiques

Dans cette dernière section, on va introduire quelques préliminaires spécifiques qui serviront lors de la construction du modèle de μLJ . Tout d'abord, on va commencer par rappeler les notions d'*algèbre initiale* et de *coalgèbre terminale* d'un endofoncteur, structures catégoriques largement habituelles pour ce qui est de la modélisation de l'induction et de la coinduction [JR97]. L'autre partie de cette section rappellera quelques propriétés de relèvements de foncteurs par l'intermédiaire de lois distributives.

1.3.1 Algèbres initiales et coalgèbres finales

Usuellement, la notion d'algèbre est rattachée à celle de *monade* [Mog91]. Ici, il s'agit d'une variante plus relâchée, ou on parle non pas d'algèbres d'une monade mais d'un *endofoncteur* $T : \mathcal{C} \rightarrow \mathcal{C}$ plus général.

Définition 1.3.1. *Soit \mathcal{C} une catégorie, et $T : \mathcal{C} \rightarrow \mathcal{C}$ un endofoncteur. Une T -algèbre est une paire (A, f) avec A un objet de \mathcal{C} et $f : T(A) \rightarrow A$. De même, une T -coalgèbre est une paire (A, f) avec $f : A \rightarrow T(A)$.*

Avant toute chose, notons que la forme $f : T(A) \rightarrow A$ d'une T -algèbre donne à A le statut d'*invariant* de T . Cette vision de A comme invariant peut aider à donner quelques intuitions sur les définitions qui suivent. Le lecteur intéressé pourra trouver dans [JR97] une très bonne introduction, élémentaire, des rapports entre algèbres initiales et induction, et coalgèbres terminales et coinduction. Comme d'habitude avec la définition de nouvelles structures catégoriques, il convient d'étudier les *morphismes* entre ces structures. C'est cette notion de morphisme qui permet de définir les T -algèbres initiales et T -coalgèbres terminales.

Définition 1.3.2. *Soit $T : \mathcal{C} \rightarrow \mathcal{C}$ un endofoncteur sur \mathcal{C} . Un morphisme d'algèbres $\gamma : (A_1, f_1) \rightarrow (A_2, f_2)$ est un morphisme $\gamma : A_1 \rightarrow A_2$ tel que le carré suivant commute :*

$$\begin{array}{ccc} T(A_1) & \xrightarrow{T(\gamma)} & T(A_2) \\ f_1 \downarrow & & \downarrow f_2 \\ A_1 & \xrightarrow{\gamma} & A_2 \end{array}$$

De même, un morphisme de coalgèbres $\gamma : (A_1, f_1) \rightarrow (A_2, f_2)$ est un morphisme $\gamma : A_1 \rightarrow A_2$ tel que

$$\begin{array}{ccc} A_1 & \xrightarrow{\gamma} & A_2 \\ \downarrow f_1 & & \downarrow f_2 \\ T(A_1) & \xrightarrow{T(\gamma)} & T(A_2) \end{array}$$

commute. On peut composer ces morphismes par la composition de ces diagrammes commutatifs, ce qui permet de construire, si T est un endofoncteur, la catégorie des algèbres de T et morphismes d'algèbres, et celle des coalgèbres de T et des morphismes de coalgèbres. Une algèbre initiale de T est alors un objet initial dans la catégorie des T -algèbres, et une coalgèbre terminale de T est un objet terminal dans la catégorie des T -coalgèbres. Comme d'habitude avec les propriétés universelles les algèbres initiales et coalgèbres terminales sont uniques à isomorphisme près.

Exemples. Quelques algèbres initiales et coalgèbres terminales :

- Dans la catégorie **Ens** des ensembles et fonctions, considérons l'endofoncteur $T(X) = 1 + X$, où 1 représente n'importe quel singleton. Alors l'ensemble \mathbb{N} des entiers naturels muni de la bijection canonique $f : 1 + \mathbb{N} \rightarrow \mathbb{N}$ est une algèbre initiale de T .
- Dans la catégorie **Ens**, considérons l'endofoncteur $T(X) = \mathbb{N} \times (1 + X \times X)$. Considérons alors l'ensemble \mathcal{A} des arbres binaires finis ou infinis, dont les nœuds sont étiquetés par des entiers naturels. Il existe alors un isomorphisme $f : \mathcal{A} \rightarrow \mathbb{N} \times (1 + \mathcal{A} \times \mathcal{A})$. Alors, (\mathcal{A}, f) est une coalgèbre terminale à l'endofoncteur T .

Remarque. Dans tous ces exemples, on note que le morphisme constituant l'algèbre initiale ou la coalgèbre terminale est toujours un isomorphisme. C'est en fait le cas général, comme en témoigne la proposition suivante :

Proposition 1.3.1. *Soit T un endofoncteur et (A, f) une algèbre initiale de T . Alors f est un isomorphisme. Dualement, si (A, f) est une coalgèbre terminale, f est un isomorphisme.*

Démonstration. Voir par exemple [Lam68]. □

Points fixes paramétrés. Si T est un endofoncteur disposant d'une algèbre initiale et d'une coalgèbre terminale, les définitions précédentes permettent de donner des plus petites et plus grandes solutions à des équations récursives de la forme :

$$X = T(X)$$

Dans certains cas, on veut pouvoir résoudre des systèmes d'équations récursives, comme par exemple

$$\begin{aligned} X &= S(Y, X) \\ Y &= T(X, Y) \end{aligned}$$

Par exemple, on pourrait définir (dans **Ens**) simultanément l'ensemble des arbres d'arité quelconque et celui des listes d'arbres d'arité quelconque comme étant la plus petite solution

(X, Y) de l'équation suivante :

$$\begin{aligned} X &= 1 + Y \\ Y &= 1 + X \times Y \end{aligned}$$

En programmation, les types correspondants sont appelés types inductifs/coinductifs *entrelacés* [Mat98], pour des raisons qui apparaîtront clairement au Chapitre 5. Une extension des outils ci-dessus permet de les traiter : c'est la notion d'algèbres initiales et coalgèbres terminales *paramétrées*. Si $T(X, Y)$ est un endofoncteur, une *algèbre initiale paramétrée* de T sera un foncteur $T^\mu(Y)$ satisfaisant la propriété que pour tout objet A , $T^\mu(A)$ sera une algèbre initiale de $T(-, A)$. De tels foncteurs permettent de résoudre les systèmes d'équations à deux inconnues, par exemple une solution du système ci-dessus peut être obtenue à partir d'une solution de celle-ci :

$$Y = T(S^\mu(Y), Y)$$

C'est la propriété de Bekič (une preuve dans le contexte des foncteurs et algèbres initiales peut être trouvée dans [LS81]), qu'on peut mettre en lien avec la méthode du pivot de Gauss [AN01, SP00] pour résoudre les systèmes d'équations linéaires. L'existence des algèbres initiales paramétrées et coalgèbres terminales paramétrées est assurée par la propriété suivante :

Proposition 1.3.2. *Soit $F : \mathcal{C} \times \mathcal{D} \rightarrow \mathcal{C}$ tel que pour tout objet A de \mathcal{D} il existe une algèbre initiale $(F^\mu(A), \chi_A)$ au foncteur $F(-, A)$. Alors il existe une unique façon d'étendre la collection des $F^\mu(A)$ en un foncteur de sorte que $\chi_A : F(F^\mu(A), A) \rightarrow F^\mu(A)$ soit un isomorphisme naturel. Le foncteur $F^\mu : \mathcal{D} \rightarrow \mathcal{C}$ est alors appelé une algèbre initiale paramétrée de F .*

De même, si pour tout objet A de \mathcal{D} il existe une coalgèbre terminale $(F^\nu(A), \chi_A)$ au foncteur $F(-, A)$, alors on peut étendre de façon unique $F^\nu(-)$ en un foncteur, faisant de χ_A un isomorphisme naturel. F^ν est alors appelé une coalgèbre terminale paramétrée de F .

Démonstration. Si $f : A \rightarrow A'$, on définit $F^\mu(f)$ comme étant l'unique morphisme de $F(-, A)$ -algèbre de l'algèbre initiale $(F^\mu(A), \chi_A)$ vers l'algèbre $(F^\mu(A'), F(F^\mu(A'), f); \chi_{A'})$. La functorialité de F^μ est évidente par l'initialité de $(F^\mu(A), \chi_A)$, et pour tout A , χ_A est un isomorphisme par la Proposition 1.3.1. La naturalité de χ_A est immédiate par définition de $F^\mu(f)$. Le cas de F^ν est dual. \square

1.3.2 Catégories de co-Kleisli et relèvements

On rappellera ici le concept de *comonade* sur une catégorie, et la définition attenante de *catégorie de co-Kleisli*. On verra comment la notion de *loi distributive* permet de construire un relèvement de n'importe quel endofoncteur sur la catégorie de co-Kleisli.

Comonades et catégories de co-Kleisli. Commençons par quelques définitions usuelles.

Définition 1.3.3. *Soit \mathcal{C} une catégorie. Une comonade sur \mathcal{C} est un triplet (T, δ, ϵ) , où $T : \mathcal{C} \rightarrow \mathcal{C}$ est un endofoncteur sur T , $\delta : T \xrightarrow{\bullet} TT$ et $\epsilon : T \xrightarrow{\bullet} Id$ sont des transformations naturelles vérifiant les diagrammes de cohérence suivants, pour tout $A \in \mathcal{C}_0$.*

$$\begin{array}{ccc} T(A) & \xrightarrow{\delta_A} & T^2(A) \\ \delta_A \downarrow & & \downarrow T(\delta_A) \\ T^2(A) & \xrightarrow{\delta_{T(A)}} & T^3(A) \end{array} \qquad \begin{array}{ccc} T(A) & \xrightarrow{\delta_A} & T^2(A) \\ \delta_A \downarrow & \searrow & \downarrow T(\epsilon_A) \\ T^2(A) & \xrightarrow{\epsilon_{T(A)}} & T(A) \end{array}$$

La propriété des comonades qui nous intéressera dans cette thèse, est la construction de la *catégorie de co-Kleisli* :

Définition 1.3.4. Soit \mathcal{C} une catégorie et (T, δ, ϵ) une comonade sur \mathcal{C} . On définit la catégorie de co-Kleisli de T , notée \mathcal{C}_T , par :

- Ses objets sont les objets de \mathcal{C} ;
- Ses morphismes sont définis par :

$$\mathcal{C}_T(A, B) = \mathcal{C}(T(A), B)$$

- Si $f : T(A) \rightarrow B$ et $g : T(B) \rightarrow C$, leur composée est donnée par :

$$T(A) \xrightarrow{\delta_A} T^2(A) \xrightarrow{T(f)} T(B) \xrightarrow{g} C$$

- L'identité sur A est $\epsilon_A : T(A) \rightarrow A$.

Des calculs immédiats montrent que ces données définissent bien une catégorie.

Comonade de mise en contexte. Soit \mathcal{C} une catégorie cartésienne, et $\Gamma \in \mathcal{C}_0$. Considérons alors les données suivantes :

$$\begin{aligned} T(X) &= \Gamma \times X \\ \delta_A &= \langle \pi_1, id_{\Gamma \times A} \rangle \\ \epsilon_A &= \pi_2 \end{aligned}$$

Alors il est immédiat de vérifier que (T, δ, ϵ) définit une comonade sur \mathcal{C} . Sa catégorie de co-Kleisli, notée dans cette thèse \mathcal{C}_Γ , est la catégorie des «termes dans le contexte Γ », c'est-à-dire, intuitivement, la catégorie déduite de \mathcal{C} dont les morphismes sont des termes «à variables libres» dans Γ . Le lecteur pourra d'ailleurs trouver dans [LS88] (polynomial categories) une définition équivalente de \mathcal{C}_Γ comme la catégorie des morphisme avec une variable libre, c'est-à-dire une *inconnue* $x : 1 \rightarrow \Gamma$. Puisque cette catégorie de co-Kleisli particulière sera d'une importance notable pour nous, donnons-en dorés et déjà quelques propriétés connues.

Proposition 1.3.3 (Préservation de la structure cartésienne fermée). *Si \mathcal{C} est une catégorie cartésienne fermée et $\Gamma \in (\mathcal{C}_\Gamma)_0$, alors \mathcal{C}_Γ est toujours cartésienne fermée.*

Démonstration. Voir [LS88]. □

Proposition 1.3.4 (Préservation des sommes). *Si \mathcal{C} est une catégorie cartésienne fermée, on a les propriétés suivantes de préservation des sommes :*

1. *Si \mathcal{C} est bicartésienne fermée, \mathcal{C}_Γ l'est aussi ;*
2. *Si \mathcal{C} a des coproduits faibles fonctoriels, alors \mathcal{C}_Γ aussi.*

Démonstration. La preuve de (1) peut être trouvée dans [LS88] : notons qu'on utilise de façon cruciale la distributivité $\gamma_{\Gamma, A, B}$. La même preuve fonctionne toujours pour (2) ; une des équations à montrer est légèrement plus faible. □

Lois distributives. De façon générale, une *loi distributive* d'un endofoncteur S sur un endofoncteur T est une transformation naturelle $\lambda : \overset{\bullet}{ST} \rightarrow TS$. Beck [Bec69] définit originellement les lois distributives entre monades, et on trouve dans la littérature de nombreux autres situations [CHP04]. Ici, on va s'intéresser au cas (folklorique) où S est une comonade et T est un endofoncteur quelconque, et on va voir comment une telle loi de distributivité permet de construire un relèvement de T sur \mathcal{C}_S . Mais tout d'abord, qu'est-ce qu'un relèvement ?

Définition 1.3.5 (Relèvement). *Soit \mathcal{C} une catégorie, (S, δ, ϵ) une comonade sur \mathcal{C} et T un endofoncteur de \mathcal{C} . Un relèvement de T à \mathcal{C}_S est un endofoncteur*

$$T_S : \mathcal{C}_S \rightarrow \mathcal{C}_S$$

de \mathcal{C}_S tel que pour tout $f : A \rightarrow B$ dans \mathcal{C} ,

$$T_S(\epsilon_A; f) = \epsilon_{T(A)}; T(f)$$

Autrement dit, l'endofoncteur T s'étend naturellement à la catégorie de co-Kleisli sur S , et ce de manière compatible avec son action sur \mathcal{C} . Notons qu'une des problématiques de cette thèse (Chapitre 6) sera la construction d'un relèvement simultanément de toute une classe de foncteurs interprétant les types, et vérifiant certaines conditions d'uniformité.

Pour obtenir un relèvement d'un endofoncteur T sur \mathcal{C}_S , il suffit d'avoir une loi distributive :

Définition 1.3.6. *Une loi distributive d'une comonade S sur un endofoncteur T est une transformation naturelle :*

$$\lambda : ST \overset{\bullet}{\rightarrow} TS$$

satisfaisant les diagrammes de cohérence suivants, pour $A \in \mathcal{C}_0$.

$$\begin{array}{ccc} ST(A) & \xrightarrow{\lambda_A} & TS(A) \\ \delta_{T(A)} \downarrow & & \downarrow T(\delta_A) \\ SST(A) & \xrightarrow[S(\lambda_A)]{} STS(A) \xrightarrow[\lambda_{S(A)}]{} & TSS(A) \end{array} \quad \begin{array}{ccc} ST(A) & \xrightarrow{\lambda_A} & TS(A) \\ \epsilon_{T(A)} \searrow & & \swarrow T(\epsilon_A) \\ & T(A) & \end{array}$$

Exemple. Dans une catégorie bicartésienne fermée¹, la distributivité du produit sur la somme

$$\Gamma \times (A + A) \rightarrow \Gamma \times A + \Gamma \times A$$

est une loi distributive de la comonade $\Gamma \times -$ sur l'endofoncteur $- + -$.

Proposition 1.3.5. *Soit (S, δ, ϵ) une comonade sur une catégorie \mathcal{C} et T un endofoncteur sur \mathcal{C} . On a alors un isomorphisme entre les lois distributives $\lambda : ST \overset{\bullet}{\rightarrow} TS$ de S sur T et les relèvements de T sur la catégorie de co-Kleisli \mathcal{C}_S .*

Démonstration. Si $f : SA \rightarrow B$ est un morphisme dans \mathcal{C}_S , le morphisme $T_S(f)$ est défini par :

$$ST(A) \xrightarrow{\lambda_A} TS(A) \xrightarrow{T(f)} T(B)$$

¹Ou cartésienne fermée à coproduits faibles fonctoriels.

Inversement, si T admet un relèvement T_S dans \mathcal{C}_S , on définit :

$$\lambda_A = T_S(id_{S(A)})$$

Un calcul direct assure que c'est bien une loi distributive de la comonade S sur l'endofoncteur T , et que ces deux constructions sont inverses l'une de l'autre. \square

Ainsi il est équivalent d'étudier les relèvements de T sur \mathcal{C}_S et les lois distributives de S sur T .

Chapitre 2

Sémantique de Jeux

Sommaire

2.1 Jeux et stratégies	35
2.1.1 Coups, arènes	35
2.1.2 Constructions sur les arènes	37
2.1.3 Parties légales	39
2.1.4 Stratégies et composition	40
2.2 Une catégorie cartésienne fermée	44
2.2.1 Fils et stratégies filaires	44
2.2.2 Vue, visibilité	46
2.2.3 Correspondance de Galois et saturation	50
2.2.4 Innocence	51
2.2.5 Structure cartésienne fermée	53
2.2.6 Coproduits faibles fonctoriels	54
2.3 Vers la complétude	57
2.3.1 Bon parenthésage	57
2.3.2 Définissabilité	58
2.4 Préliminaires spécifiques	60
2.4.1 Types récursifs à la McCusker	60
2.4.2 Isomorphismes en sémantique de jeux	61

Avant de rentrer dans le vif du sujet, on va essayer de retracer un bref historique des idées ayant contribué à amener à la sémantique de jeux «moderne» présentée ici. Avertissons tout de même le lecteur que l'histoire de la sémantique de jeux comporte de très nombreux acteurs et un certain nombre de redondances, aussi cette vision sera nécessairement partielle et incomplète.

La première inspiration a pour origine la logique mathématique, et provient de travaux proposés par Lorenzen [Lor61] au début des années 1960. L'objectif était de donner aux preuves de la logique intuitionniste une sémantique moins statique que celle, par exemple, des modèles de Kripke, afin de donner une explication «finitaire» des théorèmes de cohérence, et d'étudier les propriétés de constructivité. Le cadre était le suivant : deux personnes, qu'on appellera Joueur et Opposant, se disputent concernant la validité d'une formule logique F . Joueur tente de valider F , alors qu'Opposant tente de la réfuter. Par exemple si F est une conjonction $F_1 \wedge F_2$, un coup d'Opposant sera un choix d'une sous-formule F_1 ou F_2 qu'il souhaitera attaquer.

De même, si F est une disjonction $F_1 \vee F_2$, Joueur pourra choisir une sous-formule F_1 ou F_2 qu'il souhaitera défendre. Ces premiers travaux servirent plus tard de base à Coquand [Coq95] pour réinterpréter en termes de jeux la preuve de cohérence de Gentzen de l'arithmétique de Peano. À la connaissance de l'auteur, c'est là que les *pointeurs* font pour la première fois leur apparition, permettant aux joueurs de revenir sur des positions précédentes du jeu et ainsi d'adapter leur comportement en fonction de celui de leur adversaire. Citons également les travaux de Blass [Bla72, Bla92], qui donna entre autres un premier modèle de jeux de la logique linéaire [Gir87]. Dans tous les travaux précédemment cités, l'objectif était d'obtenir la *complétude*, c'est-à-dire de caractériser la prouvabilité d'une formule logique par l'existence d'une stratégie gagnante sur le jeu correspondant. En particulier, une notion de composition entre stratégies était définie pour interpréter le *modus ponens* ou la coupure, mais il n'était pas nécessaire que cette composition soit *associative*, ce que d'ailleurs elle n'était pas [Abr03].

La deuxième inspiration est celle de la sémantique dénotationnelle. L'objectif était cette fois de donner une représentation abstraite des programmes qui soit invariante par réduction. Le problème directeur était celui de la *pleine adéquation*, initié par les travaux de Milner [Mil77] et Plotkin [Plo77] : il s'agissait de caractériser abstraitement la nature des programmes séquentiels. En termes plus précis, on voulait un modèle du langage fonctionnel minimal PCF qui caractérise exactement l'*équivalence observationnelle*, c'est-à-dire qui égalise les termes indistinguables dans la syntaxe. Cette motivation provoqua l'apparition de nouvelles sémantiques de jeux. Cette fois, l'objectif de donner une interprétation invariante par réduction poussait à rechercher des *catégories* de jeux (les syntaxes étudiées étant confluentes), c'est à dire qu'on voulait que la composition soit associative. Les premières sémantiques de jeux visaient la *logique linéaire* [AJ94, HO92], et s'inspiraient des jeux de Blass, de la Géométrie de l'Interaction [Gir89] ainsi que de la construction plus ancienne par Joyal d'une catégorie à partir des jeux de Conway [Joy77]. Puis vinrent deux réponses indépendantes au problème de pleine adéquation. L'une est le modèle communément appelé AJM (pour Abramsky, Jagadeesan et Malacaria) [AJM00]. Le second est appelé HO¹ (pour Hyland et Ong) [HO00]. Notons que dans ce second modèle, on retrouve les pointeurs de [Coq95]. Ces deux modèles de jeux pleinement adéquats pour PCF furent ensuite adaptés pour fournir des modèles de très nombreux aspects des langages de programmation : citons le non déterminisme [HM99], le contrôle [Lai97], les références [AM97, AHM98], les types récursifs [McC96], ou encore les exceptions [Lai01]. Ils furent également adaptés en des modèles de logique, citons par exemple la logique polarisée [Lau04] et la logique de second ordre [Hug97, Lat08].

C'est dans la lignée du modèle HO/N pour PCF que s'inscrivent les définitions de jeux données ici. Plus précisément, on s'appuie sur les présentations qu'en ont faites McCusker [McC96] et Harmer [Har99] dans leurs thèses. Dans une première section, on va donner la définition d'arène. Les arènes sont les représentations abstraites des formules, et chacune d'entre elles génère un ensemble de *parties légales* qui ne sont rien d'autre que les chemins avec *backtracks*, sur l'arène vue comme un graphe. On donnera aussi les définitions de stratégie et de composition, et on montrera qu'on construit ainsi une catégorie de jeux et stratégies. Dans la deuxième section, on donnera les constructions nécessaires pour construire une catégorie cartésienne fermée, et donc un modèle du λ -calcul. On montrera dans une troisième section comment les structures obtenues se rapprochent de la syntaxe, au point d'avoir un résultat de définissabilité. Finalement, on introduira quelques préliminaires qui nous seront nécessaires

¹Ce second modèle fut aussi découvert indépendamment par Nickau [Nic94], aussi il conviendrait de l'appeler HO/N.

lors du développement de cette thèse.

2.1 Jeux et stratégies

2.1.1 Coups, arènes

Fixons un ensemble Σ d'*atomes*, et notons \mathbb{N}^* le monoïde libre sur l'ensemble \mathbb{N} des entiers naturels. Dans la suite, on appellera *adresse* tout $\xi \in \mathbb{N}^*$. La suite vide (ici l'adresse vide) sera désignée par $\langle \rangle$, et l'ajout d'un entier n au début d'une adresse ξ sera noté $n \cdot \xi$. Par abus de notation, on notera également $\xi_1 \cdot \xi_2$ l'adresse ξ_2 préfixée par ξ_1 .

Définition 2.1.1. *Un coup est un couple (ξ, a) où $\xi \in \mathbb{N}^*$ est une adresse et $a \in \Sigma$ est un atome. Par abus de notation, le couple (ξ, a) sera noté $\xi \cdot a$. Quand cela ne générera pas d'ambiguïté (c'est-à-dire quand le contexte désignera une occurrence particulière d'un atome dans l'arène), on assimilera $\xi \cdot a$ et l'atome correspondant a . L'ensemble des coups est noté M .*

Remarque. Considérer l'adresse n'est pas usuel en sémantique de jeux, alors qu'elle y est omniprésente : on la voit notamment dans la définition des constructions sur les arènes, où l'emploi de sommes disjointes donne à chaque coup une forme du type $in_1(in_2(\dots(in_1(\dots))))$. Certaines opérations construites dans cette thèse dépendent de l'accès à l'adresse offert par les présentes définitions, notamment la substitution d'arènes (Chapitre 7). Les notations utilisées ici pour l'adresse proviennent de la *ludique* [Gir01] de Girard, où elles ont à peu près le même rôle que dans le présent ouvrage : celui de *localiser les atomes*².

Définition 2.1.2. *Une arène est un quadruplet $A = (M_A, \lambda_A, \vdash_A, I_A)$, où :*

- M_A est un ensemble de coups ;
- $\lambda_A : M_A \rightarrow \{O, P\} \times \{Q, A\}$ est la fonction de polarité. On notera λ_A^{OP} la projection de λ_A sur $\{O, P\}$ et λ_A^{QA} sa projection sur $\{Q, A\}$;
- $\vdash_A : M_A \times M_A$ est la relation de justification ;
- I_A est l'ensemble des coups initiaux.

Ces données doivent vérifier les conditions suivantes :

- *Initialité* : si $m \in I_A$, alors $\lambda_A(m) = OQ$;
- *Alternance* : si $m \vdash_A n$, alors $\lambda_A^{OP}(m) \neq \lambda_A^{OP}(n)$;
- *Question/Réponse* : si $m \vdash_A n$ et $\lambda_A^{QA}(n) = A$, alors $\lambda_A^{QA}(m) = Q$.

Terminologie. Introduisons la terminologie usuelle en sémantique de jeux pour parler des arènes. Si $m \vdash_A n$, on dit que m justifie n . Si $\lambda_A^{OP}(m) = O$ (resp. P), on dit que m est un coup Opposant (resp. Joueur³). Si $\lambda_A^{QA}(m) = Q$ (resp. A), on dit que m est une Question (resp. une Réponse⁴). On définit sur $\{O, P\}$ une involution par $\overline{O} = P$ et $\overline{P} = O$, cette involution s'étend à λ_A^{OP} point par point, on note alors $\overline{\lambda}_A = \langle \overline{\lambda}_A^{OP}, \lambda_A^{QA} \rangle$.

²À quelques détails techniques près, l'adresse définit un plongement de l'arène dans l'*arène universelle* réflexive utilisée pour représenter le λ -calcul non typé [FH02].

³Les lettres P et O viennent de l'anglais «Player» et «Opponent».

⁴Les lettres Q et A viennent de l'anglais «Question» et «Answer».

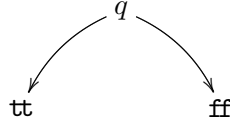


FIG. 2.1 – Représentation en graphe de l'arène des booléens.

Graphes générés. Notons qu'on peut canoniquement associer à toute arène A une structure de *graphe orienté enraciné bipartite* : les sommets sont les coups, les arêtes sont les points de \vdash_A , et la fonction de polarité assurant le caractère bipartite est λ_A^{OP} . Comme d'habitude dans la littérature on représentera souvent les arènes par cette structure de graphe dans laquelle on omettra souvent toute information sur les adresses. Par exemple, l'arène des *booléens* que l'on peut définir par :

$$\begin{aligned}
 M_{\mathbb{B}} &= \{ \langle \rangle \cdot q, 0 \cdot \mathbf{tt}, 1 \cdot \mathbf{ff} \} \\
 \vdash_{\mathbb{B}} &= \{ (\langle \rangle \cdot q, 0 \cdot \mathbf{tt}), (\langle \rangle \cdot q, 1 \cdot \mathbf{ff}) \} \\
 \lambda_{\mathbb{B}} &= \begin{cases} \langle \rangle \cdot q & \mapsto OQ \\ 0 \cdot \mathbf{tt} & \mapsto PA \\ 1 \cdot \mathbf{ff} & \mapsto PA \end{cases} \\
 I_{\mathbb{B}} &= \{ \langle \rangle \cdot q \}
 \end{aligned}$$

est représentée sous forme de graphe dans la Figure 2.1.

Relocalisation. La représentation en graphe revient à oublier l'adresse, à ne garder que les atomes et la relation de justification. De façon plus subtile il est possible de *relocaliser* une arène, c'est-à-dire de la déplacer dans l'espace physique des adresses.

Définition 2.1.3. Soit M un ensemble de coups. On définit l'ensemble des adresses de M par :

$$adr(M) = \{ \xi \mid \xi \cdot a \in M \}$$

Une relocalisation de M est alors une fonction $\kappa : adr(M) \rightarrow \mathbb{N}^*$ injective et qui préserve l'ordre préfixe sur les adresses. On définit alors M relocalisé par :

$$\kappa \cdot M = \{ \kappa(\xi) \cdot a \mid \xi \cdot a \in M \}$$

De même, si A est une arène, on définit $adr(A) = adr(M_A)$ et on définit une relocalisation de A comme une relocalisation de M_A . Si κ est une telle relocalisation, on définit A relocalisée, notée $\kappa \cdot A$, par :

$$\begin{aligned}
 M_{\kappa \cdot A} &= \kappa \cdot M_A \\
 \vdash_{\kappa \cdot A} &= \{ (\kappa(\xi_1) \cdot a, \kappa(\xi_2) \cdot b) \mid \xi_1 \cdot a \vdash_A \xi_2 \cdot b \} \\
 \lambda_{\kappa \cdot A}(\kappa(\xi) \cdot a) &= \lambda_A(\xi \cdot a) \\
 I_{\kappa \cdot A} &= \kappa \cdot I_A
 \end{aligned}$$

Il est immédiat que $\kappa \cdot A$ est toujours une arène, puisqu'on ne change pas l'égalité entre coups, à la relation de justification et à la fonction de polarité.

Citons notamment le cas le plus fréquent de relocalisation : on a une arène A et une adresse ξ et on veut relocaliser A à l'adresse ξ , c'est-à-dire préfixer toutes les adresses de A par ξ . Dans ce cas, on applique la définition ci-dessus avec la relocalisation :

$$\kappa : \xi' \mapsto \xi \cdot \xi'$$

Par abus de notation, on garde la notation ξ pour cette relocalisation, ainsi si ξ est une adresse, $\xi \cdot A$ désignera le résultat de l'opération de relocalisation avec κ donnée ci-dessus.

2.1.2 Constructions sur les arènes

Dans cette sous-section, on passe en revue les constructions fréquentes sur les arènes. Ces constructions sont similaires à celles qu'on peut trouver dans la littérature, mais la minupulation explicite des adresses permet de les rendre *localisées*.

Définition 2.1.4. Soient A et B deux arènes et $i \neq j$ des entiers. On définit l'arène produit $A \times_{i,j} B$ localisée sur i, j par les éléments suivants :

$$\begin{aligned} M_{A \times_{i,j} B} &= i \cdot M_A \uplus j \cdot M_B \\ \lambda_{A \times_{i,j} B} &= [i \cdot \lambda_A, j \cdot \lambda_B] \\ \vdash_{A \times_{i,j} B} &= i \cdot \vdash_A \uplus j \cdot \vdash_B \\ I_{A \times_{i,j} B} &= i \cdot I_A \uplus j \cdot I_B \end{aligned}$$

Par défaut, on fixe une version du produit à localité non explicite :

$$A \times B = A \times_{0,1} B$$

Définition 2.1.5. Soient A et B deux arènes et $i \neq j$ des entiers. On définit l'arène flèche $A \Rightarrow_{i,j} B$ localisée sur i, j par les éléments suivants :

$$\begin{aligned} M_{A \Rightarrow_{i,j} B} &= i \cdot M_A \uplus j \cdot M_B \\ \lambda_{A \Rightarrow_{i,j} B} &= [\overline{i \cdot \lambda_A}, j \cdot \lambda_B] \\ \vdash_{A \Rightarrow_{i,j} B} &= i \cdot \vdash_A \uplus j \cdot \vdash_B \uplus (i \cdot I_A \times j \cdot I_B) \\ I_{A \Rightarrow_{i,j} B} &= j \cdot I_B \end{aligned}$$

Par défaut, on fixe une version de la flèche à localité non explicite :

$$(A \Rightarrow B) = (A \Rightarrow_{0,1} B)$$

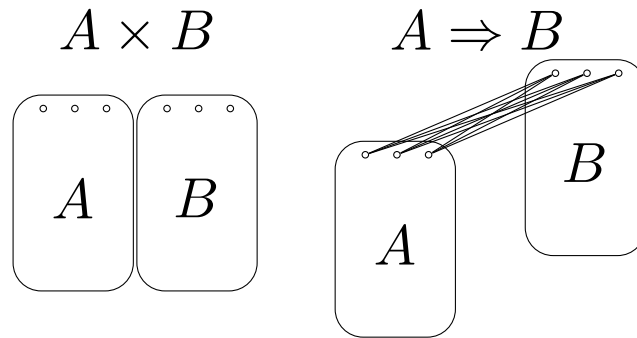
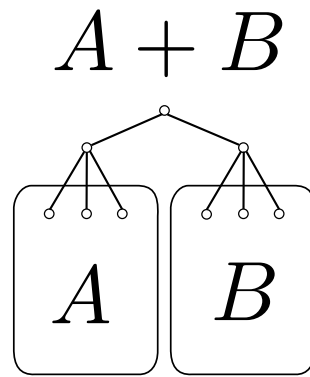
Graphes engendrés. On donne en Figure 2.2 les graphes engendrés par les constructions de produit et de flèche sur les arènes.

Définition 2.1.6. Soient A et B des arènes et $i \neq j$ des entiers. On définit l'arène somme $A +_{i,j} B$ localisée sur i, j par les éléments suivants :

$$\begin{aligned} M_{A +_{i,j} B} &= \{q, L, R\} \uplus i \cdot M_A \uplus j \cdot M_B \\ \lambda_{A +_{i,j} B} &= [q \mapsto OQ, L \mapsto PA, R \mapsto PA, i \cdot \lambda_A, j \cdot \lambda_B] \\ \vdash_{A +_{i,j} B} &= \{(q, L), (q, R)\} \uplus (\{L\} \times i \cdot I_A) \uplus (\{R\} \times j \cdot I_B) \uplus i \cdot \vdash_A \uplus j \cdot \vdash_B \\ I_{A +_{i,j} B} &= \{q\} \end{aligned}$$

On a fixé une fois pour toutes des atomes $q, L, R \in \Sigma$ réservés à cette construction de somme. Comme pour le produit, on fixe une convention pour une somme à localité non explicite par :

$$A + B = A +_{0,1} B$$

FIG. 2.2 – Graphes engendrés par $A \times B$ et $A \Rightarrow B$.FIG. 2.3 – Graphe engendré par $A + B$.

Remarque. Dans la définition ci-dessus, on n'a pas spécifié d'adresse aux atomes q, L, R : cela signifie qu'ils sont positionnés à l'adresse vide. Il serait utile de les équiper d'adresses si on avait besoin de faire des restrictions de parties aux sous-arènes commençant par L ou R mais cela n'arrivera jamais dans cette thèse.

Graphe engendré. On donne en Figure 2.3 une représentation du graphe engendré par la construction de somme.

Remarque. Il est immédiat de généraliser le produit et la somme à des versions d'arité quelconque, en enracinant chaque sous-arène à une adresse commençant par un entier n , au lieu de les localiser par défaut aux adresses 0 et 1. On utilisera occasionnellement dans cette thèse ces versions n -aires sans en détailler la définition au préalable.

Finalement, donnons la définition des *arènes unité* :

Définition 2.1.7. *L'arène 1 est l'arène dont toutes les composantes sont vides. L'arène 0 est*

donnée par les composantes suivantes :

$$\begin{aligned} M_0 &= \{\circ\} \\ \lambda_0 &= \circ \mapsto OQ \\ \vdash_0 &= \emptyset \\ I_0 &= \{\circ\} \end{aligned}$$

2.1.3 Parties légales

Maintenant qu'on a défini les arènes, il nous faut définir les parties qu'elles engendrent et leur combinatoire.

Définition 2.1.8. Soit M un ensemble de coups. Un mot à pointeurs sur M est une suite $s = s_1 \dots s_n$ de coups de M , où chaque coup s_j est équipé d'au plus un pointeur vers un coup précédent s_i (avec $i < j$). On dit alors que s_j pointe vers s_i .

Notation. Fixons quelques notations sur les mots à pointeurs. Si s est un mot à pointeurs, notons $|s|$ sa longueur. Notons \sqsubseteq l'ordre préfixe sur les mots à pointeurs : s_1 est un préfixe de s_2 si le début de s_2 coïncide avec s_1 , y compris en ce qui concerne les pointeurs. On utilisera la notation spécialisée $s_1 \sqsubseteq^O s_2$ (resp. $s_1 \sqsubseteq^P s_2$) pour préciser que s_1 termine par un coup Opposant (resp. termine par un coup Joueur). Le mot vide sera noté ϵ .

Remarque. Dans le contexte d'une partie légale, le terme «coup» est ambigu. Dans une arène A , un coup est juste un élément de M_A . Mais dans le contexte d'une partie légale, le terme «coup» désigne souvent à la fois le sommet et son pointeur associé. On explique cette subtilité par la distinction entre le «coup» qui est le sommet de l'arène, et l'acte de «jouer un coup», qui consiste à rajouter à la partie courante un élément $m \in M_A$ équipé d'un pointeur.

Terminologie. Soit s un mot à pointeurs, et supposons que s_j pointe sur s_i . Dans ce cas on dit également que s_i est le justifieur de s_j . S'il existe une séquence de pointeurs allant de s_j à s_i dans s ($i < j$), on dit que s_j est héréditairement justifié par s_i .

Définition 2.1.9. Soit A une arène. Les parties légales sur A sont les mots à pointeurs sur M_A qui sont :

- Alternés : si $i \in \{1, \dots, |s| - 1\}$, alors $\lambda_A^{OP}(s_i) \neq \lambda_A^{OP}(s_{i+1})$;
- Bien ouverts : si s_i n'a pas de pointeur associé, alors $s_i \in I_A$;
- Bien justifiés : si s_j pointe vers s_i , alors $s_i \vdash_A s_j$;

Notons \mathcal{L}_A l'ensemble des parties légales sur A .

Notation. Pour des raisons typographiques évidentes, on omettra souvent les pointeurs lorsqu'on décrira des parties légales. Par exemple on pourra désigner une partie par sa pour séparer son dernier coup du corps de la partie, sans expliciter l'éventuel pointeur associé à a . De même, si pour deux parties sa et sa' on écrit $a = a'$, il sera implicite que l'égalité tient non seulement pour les coups, mais également pour l'éventuel pointeur associé.

Restriction. Au cœur de la composition et du caractère dynamique des stratégies se trouve la notion de *restriction* d'une partie légale à une adresse donnée.

Définition 2.1.10 (Restriction). *Soit A une arène, m un coup sur A , et $\Xi = \xi_1, \dots, \xi_n$ une suite d'adresses. On dit que m est dans Ξ si $m = \xi \cdot a$, avec un des ξ_i préfixe de ξ . Si s est un mot à pointeurs sur A , on obtient la restriction de s à Ξ , notée $s_{\upharpoonright \Xi}$, en ne gardant que les coups de s dans Ξ . Si s_i et s_j sont dans Ξ et que s_j pointe vers s_i , on garde ce pointeur dans $s_{\upharpoonright \Xi}$. Si par contre s_j pointe hors de Ξ , on suit cette séquence de pointeurs jusqu'à tomber sur s_k dans Ξ , on fait alors pointer s_j vers s_k dans $s_{\upharpoonright \Xi}$. Si on ne trouve jamais de tel s_k , c'est-à-dire qu'on ne retombe jamais dans Ξ , alors s_j perd son pointeur.*

Notation. Formellement, la restriction est définie sur une séquence d'adresses $\Xi = \xi_1, \dots, \xi_n$. En pratique on l'utilisera surtout, comme c'est courant en sémantique de jeux, en donnant une sous-arène. Par exemple, si $s \in \mathcal{L}_{A \rightarrow B}$, on écrira $s_{\upharpoonright A}$ plutôt que $s_{\upharpoonright 0}$ et $s_{\upharpoonright B}$ plutôt que $s_{\upharpoonright 1}$. À chaque fois qu'on utilisera cette notation, la sous-arène à laquelle on restreint correspondra de façon non ambiguë à une adresse en particulier.

Relocalisation. Si A est une arène, $s \in \mathcal{L}_A$ et κ est une relocalisation de A , alors on définit la *relocalisation* de s associée, notée $\kappa \cdot s$, par la relocalisation simultanée de tous ses coups. Il est évident qu'on a toujours une partie légale, puisque le seul élément qui change est l'adresse et que la notion de partie légale ne dépend en aucune façon de l'adresse⁵.

2.1.4 Stratégies et composition

Stratégies. Définissons maintenant le concept de stratégie sur une arène.

Définition 2.1.11 (Stratégie). *Soit A une arène. Une stratégie sur A est un ensemble σ de parties légales finissant par un coup Joueur, qui est clos par préfixe : pour tout $s \in \sigma$, si $s' \sqsubseteq^P s$, alors $s' \in \sigma$. Si σ est une stratégie, on définit son domaine par :*

$$\text{dom}(\sigma) = \{so \mid s \in \sigma \wedge s'o \in \mathcal{L}_A\}$$

Si σ est une stratégie sur A , on note $\sigma : A$.

Il faut voir une stratégie σ comme une sorte de «dictionnaire», que Joueur consulte avant de jouer son coup. Pour chaque partie s dans le domaine de σ , il récupère un ensemble d'extensions sm possibles, et il choisit son comportement parmi elles de façon non déterministe. Notons au passage la classe des stratégies *déterministes* pour lesquelles il y a toujours au plus une possibilité, et les stratégies *totales* pour lesquelles il y a toujours au moins une possibilité.

Définition 2.1.12. *Une stratégie $\sigma : A$ est déterministe si pour tout $sa \in \text{dom}(\sigma)$, si $sab \in \sigma$ et $sab' \in \sigma$, alors $b = b'$. Une stratégie est totale si pour tout $sa \in \text{dom}(\sigma)$ il existe un coup b tel que $sab \in \sigma$.*

D'un point de vue calculatoire, il est clair que le déterminisme correspond aux programmes déterministes. Ce qui est par contre moins clair, c'est que la totalité correspond aux programmes normalisants. On n'en dit pas plus pour le moment, étant donné qu'on aura de multiples occasions d'y revenir plus tard. Notons ce lemme, véritable couteau suisse de l'étude des stratégies totales.

⁵Sauf pour ce qui est de l'égalité sur les coups, mais l'injectivité des relocalisations assure qu'elle ne change pas.

Lemme 2.1.1. *Soient $\sigma : A$ et $\tau : A$ deux stratégies déterministes et totales, telles que $\sigma \subseteq \tau$. Alors $\sigma = \tau$.*

Démonstration. Soit $s \in \tau$, considérons le plus long préfixe s' de s qui est dans σ . Si $s' = s$, alors c'est fini. Sinon, il existe un coup Opposant a tel que $s'a \sqsubseteq^O s$. Mais σ est déterministe et totale, donc il existe un unique coup b (et un pointeur associé) tel que $s'ab \in \sigma$. Or $\sigma \subseteq \tau$, donc $s'ab \in \tau$, de quoi on déduit immédiatement que $s'ab \sqsubseteq s$ par déterminisme de τ : c'est absurde, donc $s' = s$ et $s \in \sigma$. \square

Relocalisation. Si on a $\sigma : A$ et que κ est une relocalisation de A , on définit la relocalisation de σ associée, notée $\kappa \cdot \sigma$, par la relocalisation simultanée de toutes ses parties. Par injectivité de κ , $\kappa \cdot \sigma$ garde toutes les propriétés de σ (déterminisme, totalité). Un cas fréquent d'usage de cette définition est celui où on a $\sigma : A \Rightarrow B$ qu'on veut voir comme une stratégie sur $A \Rightarrow_{i,j} B$ ou inversement. Dans ce cas on note $\sigma^{i,j} : A \Rightarrow_{i,j} B$ la stratégie obtenue par la relocalisation évidente.

Composition. Il est maintenant temps de transformer ces stratégies en objets dynamiques, c'est-à-dire de définir leur composition. Dans le cadre présent, l'idée est la suivante : à partir de $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$, l'objectif est de définir une nouvelle stratégie $\sigma; \tau : A \Rightarrow C$. Cette opération est accomplie par le mécanisme habituel d'*interaction parallèle* et de *masquage*. L'idée est de faire communiquer σ et τ à travers B , puis de masquer la partie centrale de l'interaction, afin de récupérer une stratégie sur $A \Rightarrow C$.

Définition 2.1.13 (Interaction). *Soient A, B et C des arènes. Une interaction sur A, B et C est un mot à pointeurs u sur $0 \cdot M_A, 1 \cdot M_B$ et $2 \cdot M_C$ tel que $u \upharpoonright_{A,B} \in \mathcal{L}_{A \Rightarrow_{0,1} B}$, $u \upharpoonright_{B,C} \in \mathcal{L}_{B \Rightarrow_{1,2} C}$ et $u \upharpoonright_{A,C} \in \mathcal{L}_{A \Rightarrow_{0,2} C}$. Notons $I(A, B, C)$ l'ensemble des interactions sur A, B et C .*

Remarque. Rendre les adresses explicites permet de clarifier quelques subtilités présentes dans la version traditionnelle de cette définition, souvent laissées de côté dans la littérature. En particulier, $I(A, B, C)$ est en général défini comme l'ensemble des mots à pointeurs sur A, B et C dont les restrictions respectives à $A \Rightarrow B$, $B \Rightarrow C$ et $A \Rightarrow C$ sont légales. Mais si la restriction est neutre (*i.e.* n'effectue aucun réadressage) et que l'ensemble des coups de $A \Rightarrow B$ est obtenu par somme disjointe, comment cet ensemble peut-il être non vide ? En particulier, un coup sur B doit-il être de la forme $in_1(m)$ ou $in_2(m)$? Certes, ces problèmes ne sont pas fondamentaux : en général, la restriction effectue implicitement la relocalisation nécessaire. Cependant, on a choisi de tirer ici parti de la localité pour en donner une version non ambiguë.

Définition 2.1.14. *Soient $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies. On définit d'abord l'interaction entre σ et τ par :*

$$\sigma \parallel \tau = \{u \in I(A, B, C) \mid u \upharpoonright_{A,B} \in \sigma^{0,1} \wedge u \upharpoonright_{B,C} \in \tau^{1,2}\}$$

On en déduit alors :

$$(\sigma; \tau)^{0,2} = \{u \upharpoonright_{A,C} \mid u \in \sigma \parallel \tau\}$$

On obtient alors $(\sigma; \tau) : A \Rightarrow C$ par relocalisation. Notons que $s \in (\sigma; \tau)^{0,2}$ si et seulement si il existe $u \in \sigma \parallel \tau$ tel que $u \upharpoonright_{A,C} = s$. Dans ce cas, on dit que u est un témoin pour s .

Remarque. L'idée ici est de donner des bases saines à la sémantique de jeux, pas de s'encombrer avec toute une terminologie supplémentaire. En conséquence, on va maintenant passer sous silence les phases de relocalisation (sauf si elles ne sont pas évidentes), et tâcher de retrouver le langage usuel de la littérature.

Identités. Si A est une arène, il est assez facile d'exhiber une stratégie jouant le rôle de l'identité sur A . Son comportement est simple à décrire : cette stratégie se contente de recopier le comportement de son Opposant. Citons l'image, certes très connue mais toujours parlante, d'un joueur d'échecs médiocre jouant simultanément les blancs contre Garry Kasparov et les noirs contre Anatoli Karpov. Ce joueur peut facilement s'assurer la victoire contre au moins un de ces adversaires en se contentant de recopier les coups et de les faire jouer les deux joueurs l'un contre l'autre. C'est ainsi que procède l'identité en sémantique de jeux. Dans la suite, on emploiera parfois le qualificatif d'origine anglophone de «stratégie copycat» pour désigner une stratégie qui, comme l'identité, se contente de recopier le comportement de son opposant.

Définition 2.1.15. *Soit A une arène. Dans la définition ci-dessous, on utilisera des indices A_1, A_2 pour distinguer les différentes copies de A . La stratégie identité sur cette arène, notée id_A , est définie par :*

$$id_A = \{s \in \mathcal{L}_{A_1 \Rightarrow A_2} \mid \forall s' \sqsubseteq^P s, s'_{\uparrow A_1} = s'_{\uparrow A_2}\}$$

Lemme 2.1.2. *Soit $\sigma : A \Rightarrow B$ une stratégie. Alors :*

$$\begin{aligned} id_A; \sigma &= \sigma \\ \sigma; id_B &= \sigma \end{aligned}$$

Démonstration. Montrons la première équation, la deuxième est similaire. Soit $s \in id_A; \sigma$. Par définition de la composition, il existe un témoin $u \in id_A \parallel \sigma$ pour s . Mais par définition de id_A , $u_{\uparrow A_1} = u_{\uparrow A_2}$, donc $s = u_{\uparrow A_1, B} = u_{\uparrow A_2, B} \in \sigma$. Si $s \in \sigma$, on construit par induction immédiate un témoin pour s dans $id_A \parallel \sigma$, de sorte qu'on a bien $s \in id_A; \sigma$. \square

Associativité. Avant d'avoir une catégorie d'arènes et stratégies, il reste à montrer l'associativité de la composition. C'est la première difficulté lors de la construction d'un modèle de jeux. Cette difficulté est généralement résolue par la notion d'*interaction quaternaire* (généralisant la notion d'interaction à quatre arènes) et par le lemme de *zipping*⁶.

Définition 2.1.16. *Soient A, B, C et D quatre arènes. On définit une interaction quaternaire $u \in I(A, B, C, D)$ comme un mot à pointeurs sur A, B, C et D tel que :*

$$\begin{aligned} u_{\uparrow A, B} &\in \mathcal{L}_{A \Rightarrow B} \\ u_{\uparrow B, C} &\in \mathcal{L}_{B \Rightarrow C} \\ u_{\uparrow C, D} &\in \mathcal{L}_{C \Rightarrow D} \\ u_{\uparrow A, D} &\in \mathcal{L}_{A \Rightarrow D} \end{aligned}$$

⁶On a jugé préférable de ne pas traduire ce genre de vocabulaire technique, qui est utilisé tel quel en français dans la communauté.

Notons qu'on pourrait directement généraliser la notion d'interaction à n'importe quel nombre fini d'arènes. Ce ne sera pas nécessaire dans le présent développement, aussi nous nous en abstenons. L'intérêt de cette définition est de pouvoir démontrer et prouver le *lemme de zipping*, clé pour obtenir l'associativité.

Lemme 2.1.3 (Zipping). *Soient $u \in I(A, C, D)$ et $v \in I(A, B, C)$ deux interactions coïncidant sur A, C , c'est-à-dire que $u_{\uparrow A, C} = v_{\uparrow A, C}$. Alors il existe un unique $w \in I(A, B, C, D)$ tel que $w_{\uparrow A, C, D} = u$ et $w_{\uparrow A, B, C} = v$. On a également le résultat symétrique, permettant de construire un unique $w \in I(A, B, C, D)$ à partir de $u \in I(A, B, D)$ et $v \in I(B, C, D)$.*

Idée de preuve. Contentons-nous d'examiner le premier cas, qu'on montre par induction simultanée sur u et v . Si le dernier coup est dans A, C on sait par hypothèse que u et v s'accordent sur ce coup, on le rajoute au w donné par hypothèse d'induction : les tailles de u et v diminuent strictement lors de l'appel récursif. Sinon, le dernier coup est dans B ou D , donc est soit dans u soit dans v . Dans ce cas on place ce dernier coup à la fin du w donné par hypothèse d'induction, et la taille d'au moins un parmi u et v a diminué strictement. Si u et v sont vides, c'est fini : on obtient bien le témoin quaternaire voulu. Si w' est un autre témoin quaternaire satisfaisant les hypothèses du lemme alors $w = w'$; dans le cas contraire il suffit de considérer le premier coup qui les distingue, et une analyse des cas montre que ce désaccord apparaît nécessairement sur A, B, D ou A, C, D . \square

Proposition 2.1.4 (Associativité). *Soient $\sigma : A \Rightarrow B$, $\tau : B \Rightarrow C$ et $\delta : C \Rightarrow D$. Alors :*

$$(\sigma; \tau); \delta = \sigma; (\tau; \delta)$$

Démonstration. Soit $s \in (\sigma; \tau); \delta$. Par définition de la composition, s un témoin $u \in (\sigma; \tau) \parallel \delta$. Encore par définition de la composition, $u_{A, C}$ admet un témoin $v \in \sigma \parallel \tau$. Mais par le lemme de zipping, il existe alors un unique témoin $w \in I(A, B, C, D)$ coïncidant avec u et v . On veut alors montrer que $w_{\uparrow B, C, D} \in \tau \parallel \delta$; le seul point non trivial est qu'il s'agit bien d'une interaction. Pour ça, il suffit de montrer que $w_{\uparrow B, D} \in \mathcal{L}_{B \Rightarrow D}$. Trois conditions sont à vérifier :

- Alternance : On utilise le fait que $w \in I(A, B, C, D)$, en particulier w est alterné sur toutes ses composantes parmi $A \Rightarrow B$, $B \Rightarrow C$, $C \Rightarrow D$ et $A \Rightarrow D$. Cette contrainte permet d'assurer la *localité du transfert*, c'est-à-dire que si le dernier coup est sur B , alors on doit jouer sur B avant d'atteindre C et réciproquement ; ainsi on a alternance. Ce raisonnement est très bien explicité dans [Har07].
- Ouverture, justification : trivial.

Ainsi $w_{\uparrow B, C, D} \in \tau \parallel \delta$, d'où on déduit que $w_{\uparrow A, B, D}$ est dans $\sigma \parallel (\tau; \delta)$, et par construction c'est bien un témoin pour s donc $s \in \sigma; (\tau; \delta)$. L'autre inclusion se montre de la même façon. \square

On a donc à ce point une catégorie **Str** d'arènes et de stratégies. Sans condition supplémentaire sur les stratégies, **Str** a déjà la structure assez riche suivante.

Proposition 2.1.5. *Les constructeurs d'arènes \times et \Rightarrow peuvent être étendus de façon canonique en des foncteurs, de sorte que pour toute arène A , le foncteur $A \times -$ soit adjoint à gauche au foncteur $A \Rightarrow -$. Ces foncteurs donnent alors à **Str** une structure de catégorie symétrique monoïdale fermée.*

Démonstration. La preuve peut être trouvée dans [Har07], ou [McC96]. \square

Ainsi, cette catégorie suffit d'ores et déjà pour définir une interprétation correcte du fragment multiplicatif de la logique linéaire. Bien sûr, cette interprétation de MLL n'a que peu d'intérêt, étant très loin de la complétude. On va donc continuer à enrichir cette catégorie basique de jeux jusqu'à avoir une catégorie cartésienne fermée : c'est l'objectif des notions de *vue* et d'*innocence*, que nous allons bientôt introduire. Cependant, prenons le temps auparavant de discuter de la stabilité par composition des principales propriétés des stratégies introduites jusqu'à maintenant.

Déterminisme. Une des propriétés basiques qu'on a défini sur les stratégies est la propriété de déterminisme : une stratégie est déterministe lorsqu'à chaque étape elle a au plus un coup possible. Montrons ici que cette propriété est stable par composition.

Lemme 2.1.6 (Témoin unique). *Soient $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies déterministes, et $s \in \sigma; \tau$. Alors le témoin $u \in \sigma || \tau$ de s est unique, c'est-à-dire que c'est la seule interaction $u \in I(A, B, C)$ telle que $u|_{A, B} \in \sigma$, $u|_{B, C} \in \tau$ et $u|_{A, C} = s$.*

Démonstration. Si s admet deux témoins u_1 et u_2 , alors ceux-ci ne peuvent différer que dans B : c'est donc que soit σ , soit τ a eu un comportement non déterministe, impossible. \square

Proposition 2.1.7. *Soient $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies déterministes, alors $\sigma; \tau : A \Rightarrow C$ est également déterministe.*

Démonstration. Supposons qu'il existe $s \in \mathcal{L}_{A \Rightarrow C}$ tel que $s \in \sigma; \tau$ avec une extension $sa \in \mathcal{L}_{A \Rightarrow C}$ telle que $sab \in \sigma; \tau$ et $sab' \in \sigma; \tau$. Par le lemme précédent, s a un témoin unique u , qui peut être étendu en $uavb$ témoin unique pour sab , et $uav'b'$ témoin unique pour sab' . Mais par induction immédiate, et par déterminisme de σ et τ , on a nécessairement $v = v'$, donc b et b' doivent jouer dans la même composante, A ou C . Par conséquent, l'hypothèse $b \neq b'$ implique immédiatement le non-déterminisme de σ (si b, b' sont dans A) ou de τ (si b, b' sont dans C). \square

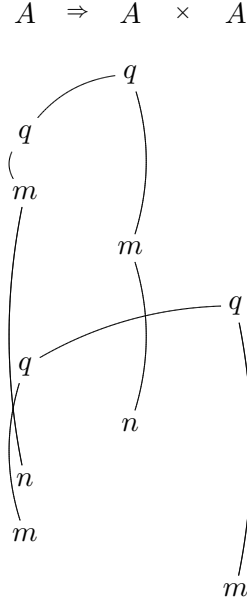
Totalité. Comme défini plus haut, une stratégie σ est totale si elle a toujours au moins un coup possible en réponse à n'importe quel coup de l'Opposant. Cette définition fait apparaître une certaine dualité entre déterminisme et totalité, ainsi on pourrait s'attendre à ce que cette propriété soit également stable par composition. Ce n'est malheureusement pas le cas, ainsi deux stratégies $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ totales peuvent donner lieu à une stratégie $\sigma; \tau$ partielle : il suffit que σ et τ se lancent dans un *dialogue infini* dans B , jamais visible de l'extérieur. C'est ce problème de préservation de la totalité par composition qui motive toute la Partie I de cette thèse.

2.2 Une catégorie cartésienne fermée

2.2.1 Fils et stratégies filaires

Bien qu'on ne s'en serve pas dans la construction de la catégorie des arènes et stratégies innocentes, il paraît nécessaire, pour faire une introduction complète aux jeux d'arène, de parler de la notion de *fil*. De façon élémentaire, un fil est une partie n'ayant qu'un unique coup initial. L'intérêt de cette notion est que des considérations sur les fils permettent de dégager la plus grande sous-catégorie de **Str** qui soit cartésienne fermée. En effet, chaque objet de **Str**

est déjà équipé d'une stratégie de *duplication* $\delta_A : A \Rightarrow A \times A$, se comportant comme l'identité sur chaque fil. Elle peut être illustrée par le schéma suivant :



De même, chaque arène est équipée d'une stratégie d'*effacement*, qui est l'unique stratégie $\eta_A : A \rightarrow 1$, ne contenant que la partie vide. Avec cette structure, on peut montrer :

Proposition 2.2.1. *Si A est une arène, le triplet (A, δ_A, η_A) est un comonoïde dans **Str**.*

Démonstration. Vérification directe. □

D'autre part on sait que si \mathcal{C} est une catégorie symétrique monoïdale fermée, la catégorie des comonoïdes cocommutatifs de \mathcal{C} et morphismes de comonoïdes est cartésienne fermée [Me108]. Ainsi apparaît naturellement la question d'isoler dans **Str** les stratégies qui se trouvent être des morphismes de comonoïde. Comme le lecteur s'en doute, c'est cette notion de *fil* qui donnera la réponse à cette question. Posons les définitions suivantes.

Définition 2.2.1. *Soit A une arène, et $s \in \mathcal{L}_A$. Si i est un coup initial de s , notons $s_{\uparrow i}$ la sous-partie de s des coups héréditairement justifiés par i . Le fil courant de s , noté $\lceil s \rceil$, est alors défini comme étant $s_{\uparrow i}$, où i est le coup initial justifiant héréditairement le dernier coup de s .*

Cette notion de fil courant donne une fonction d'information partielle sur toute partie s . On définit alors la classe des stratégies dont le comportement est entièrement caractérisé par cette information partielle.

Définition 2.2.2. *Soit $\sigma : A$ une stratégie. σ est dite filaire si elle vérifie la condition suivante :*

$$\forall s \in \mathcal{L}_A, s \in \sigma \Leftrightarrow \forall s' \sqsubseteq^P s, \lceil s' \rceil \in \sigma$$

Autrement dit, σ ne dépend que du fil courant.

Bien sûr, il existe de nombreuses formulations équivalentes de cette définition. L'élément important à retenir est qu'on peut caractériser une stratégie σ filaire par son ensemble de fils. À chaque étape, σ compare le fil courant de la partie à cet ensemble de fil, et en déduit le coup à jouer. Montrer la stabilité par composition des stratégies filaires est assez facile : il suffit de montrer que $\sigma; \tau$ peut être construite en examinant uniquement les interactions avec un seul coup initial sur C . L'intérêt des stratégies filaires réside surtout dans le théorème suivant, dont la démonstration peut être trouvée dans [Har99].

Théorème 2.2.2 (Harmer). *Les morphismes de comonoïdes dans \mathbf{Str} sont exactement les stratégies filaires.*

S'ensuit immédiatement, de par la discussion plus haut, que la catégorie des arènes et stratégies filaires est cartésienne fermée : elle fournit donc un modèle correct du λ -calcul. Le modèle fourni par cette catégorie n'est toujours pas satisfaisant, en ce sens qu'il est très loin de la complétude. De nombreuses stratégies dans cette catégorie ne viennent pas de preuves. On va donc maintenant poursuivre la préparation de l'innocence, qui fournira la catégorie cartésienne fermée satisfaisante recherchée.

2.2.2 Vue, visibilité

Pointeurs. Il y a de nombreuses façons d'interpréter les pointeurs, et chacune d'elles fournit également une intuition sur la vue. Une analogie particulièrement riche est celle du *backtrack* : durant une interaction entre deux joueurs Joueur et Opposant, l'un des deux joueurs peut décider de revenir à un état précédent. Plutôt que de représenter cela par le retour de la partie à l'un de ses préfixes, on internalise ce retour en arrière dans la syntaxe des parties par l'intermédiaire d'un pointeur. Évidemment ce pointeur doit s'accompagner de l'oubli d'une partie de l'histoire : si Opposant décide de revenir à un état précédent, le prochain coup du Joueur ne doit dépendre que de cet état restreint. C'est la *vue* qui se chargera de définir l'information partielle à laquelle les joueurs ont accès pour choisir leur prochain coup.

Vues. On donne plusieurs notions de vue, correspondant aux différents joueurs et aux différentes polarités du coup actuel.

Définition 2.2.3. *Soit s un mot à pointeurs. On définit par induction mutuelle la vue (notée $\lceil s \rfloor$) et la covue (notée $\lfloor s \lceil$) sur s :*

$$\begin{aligned} \lceil \epsilon \rfloor &= \epsilon \\ \lceil sa \rfloor &= \lfloor s \lceil a \\ \lfloor \epsilon \lceil &= \epsilon \\ \lfloor s_1 a \lceil &= a && \text{si } a \text{ n'a pas de pointeur} \\ \lfloor s_1 b s_2 a \lceil &= \lceil s_1 b \rfloor a && \text{si } a \text{ pointe sur } b \end{aligned}$$

Cette définition donne les coups de $\lceil s \rfloor$ et $\lfloor s \lceil$. Quant aux pointeurs, ils sont conservés si possible : si $s' = \lceil s \rfloor$, alors un coup s'_j pointe sur s'_i dans s si et seulement si s'_i et s'_j proviennent respectivement de $s_{\phi(i)}$ et $s_{\phi(j)}$ dans s , avec $s_{\phi(j)}$ qui pointe sur $s_{\phi(i)}$.

La *vue* est l'information partielle à laquelle a accès le joueur ayant joué le dernier coup, la *covue* l'information partielle accessible pour son adversaire. À partir de ces définitions, on retrouve les notions (familières aux lecteurs habitués à la sémantique de jeux) de *P-vue* et de *O-vue* :

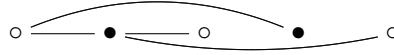
Définition 2.2.4. Soit s une partie. On définit la vue de Joueur (ou P -vue) sur s , notée $\lceil s \rceil$, par :

$$\begin{aligned} \lceil s \rceil &= \lceil s \rfloor & \text{si le dernier coup de } s \text{ est Joueur} \\ \lceil s \rceil &= \lfloor s \rceil & \text{sinon} \end{aligned}$$

De même, on définit la vue d'Opposant (ou O -vue) sur s , notée $\lfloor s \rfloor$, par :

$$\begin{aligned} \lfloor s \rfloor &= \lceil s \rfloor & \text{si le dernier coup de } s \text{ est Opposant} \\ \lfloor s \rfloor &= \lfloor s \rceil & \text{sinon} \end{aligned}$$

Visibilité. Comme noté plus haut, les pointeurs peuvent être perdus lors du calcul de la vue. En particulier, rien n'assure pour le moment que ce calcul donne des parties légales. Il est même assez facile de trouver un contre-exemple. Considérons la partie suivante, à lire de gauche à droite. Les coups blancs (\circ) sont des coups Opposant, les coups noirs (\bullet) sont des coups Joueur.



Si on calcule la vue à partir du dernier coup, celui-ci perd son pointeur. On obtient alors une partie où ce coup, non-initial, n'a pas de justificateur : cette partie ne peut donc pas être légale. Nombre de définitions dans la suite vont s'appuyer sur ces notions de vue, il faut donc cerner l'ensemble des parties sur lesquelles elles sont bien définies.

Définition 2.2.5. Soit A une arène, et $s \in \mathcal{L}_A$. On dit que s est P -visible si pour tout préfixe $s' \sqsubseteq s$, on a $\lceil s' \rceil \in \mathcal{L}_A$. Autrement dit, Joueur pointe toujours dans sa P -vue. De même, s est O -visible si pour tout préfixe $s' \sqsubseteq s$, on a $\lfloor s' \rfloor \in \mathcal{L}_A$. Autrement dit, Opposant pointe toujours dans sa O -vue. Une partie $s \in \mathcal{L}_A$ est visible si elle est à la fois P -visible et O -visible.

Conditions de saut. La première application de la condition de visibilité est qu'elle impose automatiquement les *conditions de saut*⁷, qu'on peut formuler comme suit : dans une partie $s \in \mathcal{L}_{A \times B}$, seul Opposant peut effectuer des sauts, c'est-à-dire passer de A à B . Dans une somme, ce sera l'inverse : seul Joueur pourra effectuer des sauts. Dans les deux cas, la conséquence importante est qu'une partie s pourra être restreinte à A ou B tout en préservant la légalité :

Lemme 2.2.3 (Conditions de saut). Soient A et B deux arènes, on a les propriétés suivantes :

1. Si $s \in \mathcal{L}_{A \times B}$ est P -visible, alors seul Opposant peut sauter entre A et B , ainsi on a ⁸, $s \upharpoonright_A \in \mathcal{L}_A$ et $s \upharpoonright_B \in \mathcal{L}_B$.
2. Si $s \in \mathcal{L}_{A+B}$ est un fil O -visible, alors seul Joueur peut sauter entre A et B , ainsi on a $s \upharpoonright_A \in \mathcal{L}_A$ et $s \upharpoonright_B$.
3. Si $s \in \mathcal{L}_A$ est P -visible, alors seul Opposant peut sauter entre les fils de s . Si i est un coup initial dans s , alors $s \upharpoonright_i \in \mathcal{L}_A$.
4. Si $s \in \mathcal{L}_{A+B}$ est visible, alors $s \upharpoonright_A \in \mathcal{L}_A$ et $s \upharpoonright_B \in \mathcal{L}_B$.

⁷Appelées *switching conditions* en anglais.

⁸On passe sous silence le réadressage immédiat.

Démonstration. Dans chaque cas, la seule chose à montrer est la condition d'alternance. Si s est une P -vue sur $A \times B$, on montre par induction immédiate sur s qu'elle est entièrement incluse soit dans A , soit dans B . Donc la condition de P -visibilité implique que Joueur ne peut pas sauter. De même sur $A + B$, on montre par induction immédiate qu'une O -vue filaire est de la forme qLs_1 ou qRs_2 , avec s_1 entièrement inclus dans A ou s_2 entièrement inclus dans B . Donc la condition de O -visibilité implique qu'Opposant ne peut pas faire de saut, ainsi les deux projections sont alternées donc légales. Le troisième énoncé se prouve comme le premier, et le quatrième est une conséquence immédiate des énoncés 2 et 3. \square

Remarque. De nombreux auteurs intègrent la condition de visibilité dans la définition des parties légales. Ici, on a choisi de ne pas le faire parce qu'on exploitera (brièvement) dans cette thèse cette possibilité de non-visibilité⁹. Insistons tout de même sur le fait qu'il est très rare d'utiliser cette possibilité de comportement non visible pour interpréter le calcul, l'exception notable étant la modélisation d'un langage admettant des références générales dans [AHM98]. Toutes les stratégies considérées dans cette thèse satisferont la condition suivante :

Définition 2.2.6. *Soit A une arène, et $\sigma : A$. La stratégie σ est visible si et seulement si pour tout $s \in \sigma$, s est P -visible.*

On va montrer maintenant que la classe des stratégies visibles est stable par composition. La preuve sera intéressante d'une part parce qu'elle donnera des préliminaires techniques indispensables à la preuve à venir de préservation de l'innocence par composition, et également parce que la classe des stratégies visibles est pertinente d'un point de vue sémantique : les stratégies visibles, pas nécessairement innocentes mais déterministes correspondent à des termes d'un langage de programmation à la ALGOL [AM97], avec structures de données mutables sur les types de base. La preuve repose sur la généralisation suivante de la notion de P -vue aux interactions :

Définition 2.2.7. *Soient A, B et C trois arènes, et soit $u \in I(A, B, C)$. Dans u , on appelle un coup de l'Opposant externe soit un coup Joueur de A , soit un coup Opposant de C (autrement dit, c'est un coup Opposant de $A \Rightarrow C$). La pré-vue de u , notée $\lceil u \rceil$, est obtenue par le procédé récursif suivant. Dans cette définition, o désigne un coup de l'Opposant externe.*

$$\begin{aligned} \lceil u_1 p u_2 o \rceil &= \lceil u_1 p \rceil o && \text{si } o \text{ pointe sur } p \\ \lceil u \rceil &= i && \text{si } i \text{ est initial} \\ \lceil u p \rceil &= \lceil u \rceil p && \text{sinon} \end{aligned}$$

Le choix du terme *pré-vue* s'explique par le fait que les interactions de cette forme sont exactement celles qui donnent des P -vues (*i.e.* des parties s telles que $\lceil s \rceil = s$, autrement dit Opposant pointe toujours vers le coup précédent) sur $A \Rightarrow C$ une fois la restriction effectuée. Bien sûr, cette remarque sous-entend que si $u \in I(A, B, C)$ la pré-vue correspondante satisfait elle aussi $\lceil u \rceil \in I(A, B, C)$, ce qui est faux en général : sans hypothèse de visibilité sur $A \Rightarrow B$ et $B \Rightarrow C$, le calcul de la pré-vue peut perdre des pointeurs et donner une interaction incorrecte. Par contre, l'hypothèse de visibilité suffit pour assurer que la pré-vue est une interaction correcte, comme établi par les lemmes suivants :

⁹ En effet, pour une généralité maximale, les théorèmes de finitude de la Partie I ont été énoncés pour les parties *ultimement visibles*, c'est-à-dire visibles seulement à partir d'un certain rang.

Lemme 2.2.4. *Soit $u \in I(A, B, C)$ vérifiant l'analogie de la visibilité pour les interactions, c'est-à-dire que pour chaque $u' \sqsubseteq u$, le dernier coup de u' pointe dans $\ulcorner u \urcorner$. Soit également X une composante (parmi $A \Rightarrow B$ et $B \Rightarrow C$) du dernier coup de u ¹⁰, alors $\ulcorner u \urcorner_{\downarrow X}$ est une sous-séquence de $\ulcorner u \urcorner$, c'est-à-dire que tous les coups de $\ulcorner u \urcorner_{\downarrow X}$ sont compris dans $\ulcorner u \urcorner$.*

Démonstration. La preuve se fait par induction sur u . Si le dernier coup est un coup initial sur X , c'est évident ; ce cas comprend le cas de base de l'induction. Si le dernier coup est un coup Joueur sur X , alors $\ulcorner (u'm) \urcorner_{\downarrow X} = \ulcorner u' \urcorner_{\downarrow X} m$. Or le dernier coup de u' est toujours dans X , donc par hypothèse d'induction $\ulcorner u' \urcorner_{\downarrow X}$ est bien une sous-séquence de $\ulcorner u \urcorner$, ce qui conclut ce cas. Si le dernier coup est un coup Opposant sur X , alors u est de la forme $u_1 n u_2 m$, avec m pointant sur n . Or par hypothèse sur u le coup n est dans $\ulcorner u \urcorner$, et $\ulcorner (u_1 n u_2 m) \urcorner_{\downarrow X} = \ulcorner (u_1 n) \urcorner_{\downarrow X} m$. Mais $\ulcorner (u_1 n) \urcorner_{\downarrow X}$ est une sous-séquence de $\ulcorner u_1 n \urcorner$ par hypothèse d'induction, donc $\ulcorner u \urcorner_{\downarrow X}$ est bien une sous-séquence de $\ulcorner u \urcorner$. \square

Lemme 2.2.5. *Soit $u \in I(A, B, C)$ telle que $u \upharpoonright_{A \Rightarrow B}$ et $u \upharpoonright_{B \Rightarrow C}$ sont P -visibles. Alors le calcul de $\ulcorner u \urcorner$ ne perd pas de pointeur, c'est-à-dire que pour tout $u' \sqsubseteq u$, le dernier coup de u' pointe dans $\ulcorner u \urcorner$. De plus, $\ulcorner u \urcorner \in I(A, B, C)$.*

Démonstration. Raisonnons par induction sur la longueur de u . Si le dernier coup de u est initial, l'affirmation est triviale ; ce cas comprend le cas de base. Si c'est un coup de l'Opposant externe, alors u est de la forme $u_1 p u_2 o$ (avec o pointant sur p), et alors $\ulcorner u \urcorner = \ulcorner u_1 p \urcorner o$: il est trivial que le pointeur de o est alors préservé. Si le dernier coup n'est pas un coup de l'Opposant externe, alors il existe une composante X parmi $A \Rightarrow B$ et $B \Rightarrow C$ telle que p est un coup Joueur dans X . Décomposons $u = u' p$, on a alors $\ulcorner u \urcorner = \ulcorner u' p \urcorner = \ulcorner u' \urcorner p$. On sait par hypothèse d'induction que le calcul de $\ulcorner u' \urcorner$ ne perd pas de pointeur, il reste à montrer qu'on ne perd pas le pointeur de p . Pour cela, remarquons que par hypothèse d'induction, u' vérifie les hypothèses du Lemme 2.2.4, c'est-à-dire que le calcul de $\ulcorner u' \urcorner$ ne perd pas de pointeur, or le dernier coup de u' est nécessairement un coup Opposant dans la même composante X que p . On en déduit que $\ulcorner u' \urcorner_{\downarrow X}$ est une sous-séquence de $\ulcorner u' \urcorner$, or le pointeur de p est dans $\ulcorner u' \urcorner_{\downarrow X}$ par l'hypothèse de P -visibilité sur X , dont il est bien présent dans $\ulcorner u \urcorner = \ulcorner u' \urcorner p$. Le calcul de $\ulcorner u \urcorner$ ne perd donc pas de pointeur, il est alors immédiat de vérifier que $\ulcorner u \urcorner \in I(A, B, C)$. \square

Proposition 2.2.6. *Soit $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies visibles, alors $\sigma ; \tau$ est également visible.*

Démonstration. Immédiat par le lemme précédent. \square

Transitivité de la vision. On va présenter ici un lemme supplémentaire, mettant en relation les notions de vue et d'interaction. Ce paragraphe n'est pas indispensable au développement de la catégorie des arènes et stratégies innocentes, cependant le lemme présenté sera utilisé au cours de la thèse et il est riche en intuition. Dans l'énoncé qui suit, si s_i est un coup d'une partie s , on notera $s_{\leq i}$ la partie préfixe de s dont le dernier coup est s_i . Si u est une interaction et que X est une composante parmi $A \Rightarrow B$, $B \Rightarrow C$ et $A \Rightarrow C$, on dit que s_j voit s_i dans la composante X si s_i est dans $\ulcorner (s_{\leq j}) \urcorner_{\downarrow X}$.

Lemme 2.2.7. *Soit $u \in I(A, B, C)$ telle que $u \upharpoonright_{A, B}$ et $u \upharpoonright_{B, C}$ sont P -visibles. Soient u_i un coup dans A , u_j un coup dans B et u_k un coup dans C . Alors, si u_i voit u_j dans $A \Rightarrow B$ et que u_j voit u_k dans $B \Rightarrow C$, alors u_i voit u_k dans $A \Rightarrow C$.*

¹⁰Notons qu'on demande une composante et pas la composante du dernier coup de u parce que le choix n'est pas unique, notamment lorsque ce dernier coup est dans B .

Démonstration. Calculons la pré-vue de $u_{\leq i}$. Par le Lemme 2.2.5, u vérifie les hypothèses du Lemme 2.2.4. En conséquence, $\lceil (u_{\leq i}) \upharpoonright_{A,B} \rceil$ est une sous-séquence de $\lceil u_{\leq i} \rceil$. Or puisque u_i voit u_j , u_j est dans $\lceil (u_{\leq i}) \upharpoonright_{A,B} \rceil$ donc dans $\lceil u_{\leq i} \rceil$, donc $\lceil u_{\leq j} \rceil$ est préfixe de $\lceil u_{\leq i} \rceil$. On utilise maintenant le fait que u_j voit u_k : u_k est donc dans $\lceil (u_{\leq j}) \upharpoonright_{B,C} \rceil$ donc dans $\lceil u_{\leq j} \rceil$ par le Lemme 2.2.4, donc dans $\lceil u_{\leq i} \rceil$, donc u_k est bien dans $\lceil (u_{\leq i}) \upharpoonright_{A,C} \rceil$ et u_i voit u_k dans $A \Rightarrow C$. \square

En fait, cette propriété se généralise au delà de l'énoncé ci-dessus. Tout d'abord, on peut montrer qu'il s'agit d'une relation d'ordre. Ensuite, elle ne dépend pas du fait que u_i soit dans A et u_k dans C , ça peut très bien être l'inverse, ou ils peuvent être tous deux dans la même composante. On peut montrer que de façon très générale, x voit z dans $A \Rightarrow C$ s'il le voit déjà dans $A \Rightarrow B$ ou $B \Rightarrow C$, ou s'il existe un intermédiaire y tel que x voie y dans une composante X et y voie z dans une composante Y . Cet énoncé est agréable, car il donne une vision très concrète des coups appartenant à la même P -vue dans une interaction visible. Formellement, on va cependant se contenter de l'énoncé ci-dessus, le seul qui serve techniquement dans le cadre de cette thèse.

2.2.3 Correspondance de Galois et saturation

Avec l'intuition du backtrack introduite précédemment, la contrainte de visibilité s'explique aisément. Lorsqu'Opposant effectue un retour-arrière par l'intermédiaire d'un pointeur, on a expliqué que son adversaire (ici Joueur) doit oublier les coups couverts par ce pointeur. *A fortiori*, Joueur ne peut plus pointer sur ces coups couverts : il ne peut pas effectuer de retour vers un état qui «n'a plus eu lieu». La visibilité définit précisément cette restriction sur les pointeurs, même si pour le moment les stratégies peuvent encore dépendre de l'intégralité de la partie.

Il nous faut donc maintenant caractériser les stratégies qui ne dépendent réellement que de cette information partielle. Dans ce qui suit, on utilisera la notion de P -vue comme une propriété sur les parties, et non comme une fonction. Ainsi, on dira que s est une P -vue si $\lceil s \rceil = s$, ou de façon équivalente si Opposant pointe toujours sur le coup précédent dans s .

Définition 2.2.8. Si $\sigma : A$ est une stratégie, on définit l'ensemble des P -vues de σ , noté $\lceil \sigma \rceil$, par :

$$\lceil \sigma \rceil = \{ \lceil s \rceil \mid s \in \sigma \}$$

De même, si V est un ensemble de P -vues, on définit l'ensemble des traces de V par :

$$tr(V) = \{ s \in \mathcal{L}_A \mid \forall s' \sqsubseteq^P s, \lceil s' \rceil \in V \}$$

Proposition 2.2.8. Les fonctions $tr(-)$ et $\lceil - \rceil$ forment une correspondance de Galois :

$$\lceil - \rceil \dashv tr(-)$$

Démonstration. Supposons qu'on ait une stratégie $\sigma : A$ et un ensemble de P -vues V telles que $\lceil \sigma \rceil \subseteq V$. Soit alors $s \in \sigma$, et soit $s' \sqsubseteq^P s$. Nécessairement, $\lceil s' \rceil \in \lceil \sigma \rceil \subseteq V$, donc $s \in tr(V)$. De même, si on suppose $\sigma \subseteq tr(V)$, soit $v \in \lceil \sigma \rceil$. Autrement dit il existe $s \in \sigma$ tel que $v = \lceil s \rceil$. Mais par hypothèse, on a alors également $s \in tr(V)$, d'où $\lceil s \rceil = v \in V$. \square

Saturation. En particulier, la composition $tr(\ulcorner-\urcorner)$ est une fermeture. Désormais, nous noterons

$$\bar{\sigma} = tr(\ulcorner\sigma\urcorner)$$

et appellerons cette fermeture l'opération de *saturation*, ce qui conduit à la définition suivante :

Définition 2.2.9. *Une stratégie σ est dite saturée lorsque $\bar{\sigma} = \sigma$.*

Par idempotence de la fermeture, $\bar{\sigma}$ est toujours saturée pour toute stratégie σ . Par construction, le comportement d'une stratégie saturée ne dépend que de la vue courante.

Remarque. Mentionnons une propriété troublante de l'opération de saturation : elle interagit mal avec la composition, c'est-à-dire qu'on n'a pas, en général, $\overline{\sigma;\tau} = \bar{\sigma};\bar{\tau}$. En fait, on peut se rendre compte que la classe des stratégies saturées n'est pas stable par composition, c'est-à-dire qu'on peut trouver deux stratégies σ et τ saturées telles que $\sigma;\tau$ ne soit plus saturée : un tel contre-exemple apparaît par exemple dans [Har99]. Par contre, cette propriété sera vérifiée dans le cas *déterministe*, qui amène à la notion d'innocence. Notons au passage que c'est ce défaut de stabilité par composition des stratégies saturées qui fait que cette classe de stratégies n'est pas adéquate pour modéliser le non déterminisme dans les langages de programmation.

2.2.4 Innocence

Définition 2.2.10. *Soit $\sigma : A$ une stratégie. On dit que σ est innocente si et seulement si elle est saturée et déterministe.*

On va maintenant démontrer la stabilité par composition des stratégies innocentes. Ce résultat est un des points les plus techniques lors de la construction d'un modèle de jeux, et on n'en dénombre pas énormément de preuves dans la littérature. La plupart du temps, ce résultat est démontré par des raisonnements purement combinatoires sur les P -vues et les interactions, un peu comme ceux utilisés pour démontrer la préservation de la visibilité par composition. On peut par exemple trouver cette approche dans [HO00, Har99, McC96]. Récemment, un article [HHM07] a présenté une nouvelle preuve de cette propriété de préservation de l'innocence, se reposant non plus sur des raisonnements combinatoires sur les parties mais sur des mécanismes algébriques. Cette nouvelle approche est fort intéressante, mais on va garder ici l'approche usuelle. En effet, on utilise dans cette thèse bon nombre de raisonnements nécessitant une manipulation de «bas niveau» des parties et interactions, manipulations que le cadre catégorique de [HHM07] ne facilite pas. Présentons donc maintenant le lemme fondamental nécessaire à la preuve de préservation de l'innocence.

Lemme 2.2.9. *Soit $u \in I(A, B, C)$ telle que $u \upharpoonright_{A, B}$ et $u \upharpoonright_{B, C}$ sont P -visibles, soit X une composante parmi $A \Rightarrow B$ et $B \Rightarrow C$; supposons de plus que le dernier coup de u est dans la composante X . Alors :*

$$\ulcorner u \upharpoonright_X \urcorner = \ulcorner u \urcorner \upharpoonright_X \urcorner$$

Démonstration. La preuve se fait par induction sur u . Si u est un coup initial sur X , c'est trivial. Sinon, u est de la forme $u'm$. Plusieurs cas sont alors possibles :

- Si m est un coup externe (*i.e.* sur $A \Rightarrow C$), de polarité Opposant dans X . Supposons que m pointe vers un coup n , qui est forcément dans $\ulcorner u \urcorner$ par définition. On calcule :

$$\begin{aligned}
\ulcorner u \urcorner \upharpoonright_X \urcorner &= \ulcorner u_1 n u_2 m \urcorner \upharpoonright_X \urcorner \\
&= \ulcorner (\ulcorner u_1 \urcorner n m) \urcorner \upharpoonright_X \urcorner \quad (\text{def. de pré-vue}) \\
&= \ulcorner u_1 \urcorner \upharpoonright_X \urcorner n m \quad (\text{def. de } P\text{-vue}) \\
&= \ulcorner (u_1) \urcorner \upharpoonright_X \urcorner n m \quad (\text{hyp. induction}) \\
&= \ulcorner (u_1) \urcorner \upharpoonright_X \urcorner n m \urcorner \quad (\text{def. de } P\text{-vue}) \\
&= \ulcorner u \urcorner \upharpoonright_X \urcorner
\end{aligned}$$

- Si m est de polarité Joueur dans X . Le calcul est alors immédiat, m sortant librement des $\ulcorner - \urcorner$ et restrictions.

$$\begin{aligned}
\ulcorner u' m \urcorner \upharpoonright_X \urcorner &= \ulcorner u' \urcorner \upharpoonright_X \urcorner m \\
&= \ulcorner u' \urcorner \upharpoonright_X \urcorner m \quad (\text{hyp. induction}) \\
&= \ulcorner u \urcorner \upharpoonright_X \urcorner
\end{aligned}$$

- Si m est un coup interne (*i.e.* sur B), de polarité Opposant dans X . Par le Lemme 2.2.5, on sait alors que m pointe dans $\ulcorner u \urcorner$. Appellons n le justifieur de m dans $\ulcorner u \urcorner$, c'est-à-dire que $\ulcorner u \urcorner = \ulcorner u_1 \urcorner n u_2 m$ où m pointe sur n . On conclut par le calcul suivant :

$$\begin{aligned}
\ulcorner u \urcorner \upharpoonright_X \urcorner &= \ulcorner (\ulcorner u_1 \urcorner n u_2 m) \urcorner \upharpoonright_X \urcorner \\
&= \ulcorner (\ulcorner u_1 n \urcorner) \urcorner \upharpoonright_X \urcorner (u_2) \upharpoonright_X \urcorner m \urcorner \quad (\text{def. de pré-vue et restriction}) \\
&= \ulcorner u_1 n \urcorner \upharpoonright_X \urcorner m \quad (\text{def. de } P\text{-vue}) \\
&= \ulcorner (u_1 n) \urcorner \upharpoonright_X \urcorner m \quad (\text{hyp. induction}) \\
&= \ulcorner (u) \urcorner \upharpoonright_X \urcorner \quad (\text{def. de } P\text{-vue})
\end{aligned}$$

□

Proposition 2.2.10. *Soient $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies innocentes. Alors, $\sigma; \tau$ est également innocente.*

Démonstration. On sait déjà que $\sigma; \tau$ est déterministe par la Proposition 2.1.7. Il reste donc à montrer que $\sigma; \tau$ est saturée. Autrement dit, il faut monter $\overline{\sigma; \tau} = \overline{\sigma}; \overline{\tau}$, ce qui revient à l'inclusion $\overline{\sigma; \tau} \subseteq \overline{\sigma}; \overline{\tau}$. Ainsi, prenons $s \in \overline{\sigma; \tau}$. Par définition de la saturation, on sait que pour tout préfixe $s' \sqsubseteq^P s$ il existe une partie $t_{s'} \in \sigma; \tau$ telle que $\ulcorner t_{s'} \urcorner = \ulcorner s' \urcorner$. Il nous faut maintenant construire un témoin pour s dans $\overline{\sigma}; \overline{\tau}$. Raisonnons par induction sur s . Si $s = \epsilon$, l'interaction vide est un témoin. Sinon, supposons qu'on ait un témoin u pour s , construisons un témoin pour sab . Il est immédiat de vérifier que $ua \in I(A, B, C)$. Prenons $t_{sab} \in \sigma; \tau$ tel que $\ulcorner t_{sab} \urcorner = \ulcorner sab \urcorner$. En particulier, t_{sab} finit par ab , on pose donc $t_{sab} = t'ab$. Par le Lemme 2.1.6, t_{sab} a un témoin unique $v \in \sigma \parallel \tau$ de la forme $v_1 a v_2 b$. On a $\ulcorner t'ab \urcorner = \ulcorner sab \urcorner$, autrement dit $\ulcorner u \urcorner$ est également un témoin pour $\ulcorner t'ab \urcorner$. Mais par unicité du témoin (donc par déterminisme de σ et τ), on a alors $\ulcorner u \urcorner = \ulcorner v_1 \urcorner$. L'idée est maintenant de copier v_2 coup par coup sur ua , afin de générer un témoin pour sab . Notons v_2^i la chaîne composée des i premiers coups de v_2 , on va décrire un processus inductif sur i permettant de copier v_2^i sur ua de sorte que $\ulcorner v_1 a v_2^i \urcorner = \ulcorner u a v_2^i \urcorner$. Si v_2 n'est pas entièrement copié, soit X la composante du coup d'indice $i + 1$ de v_2 . Par le Lemme 2.2.9 et l'hypothèse d'induction, on a :

$$\begin{aligned}
\ulcorner (v_1 a v_2^i) \urcorner \upharpoonright_X \urcorner &= \ulcorner v_1 a v_2^i \urcorner \upharpoonright_X \urcorner \\
&= \ulcorner u a v_2^i \urcorner \upharpoonright_X \urcorner \\
&= \ulcorner (u a v_2^i) \urcorner \upharpoonright_X \urcorner
\end{aligned}$$

Ainsi, on peut étendre uav_2^i en uav_2^{i+1} avec le même pointeur que dans $v_1av_2^i$. Une vérification immédiate montre que cela définit bien une interaction légale qui satisfait l'hypothèse d'induction. Le même raisonnement permet, une fois v_2 entièrement copié, de copier b avec le même pointeur. Par construction, chaque coup de v_2 déclenche la même P -vue de σ ou τ que dans v , ainsi on a bien $uav_2b \in \sigma \parallel \tau$ par saturation de σ et τ . Finalement, on a clairement $(uav_2b)_{\uparrow A, C} = sab$ donc c'est bien un témoin pour sab , et $\sigma; \tau$ est saturée. \square

Ainsi, la condition d'*innocence* se préserve par composition. Remarquons également que les identités sont des stratégies innocentes : elles sont en fait *sans mémoire*, c'est-à-dire qu'elles ne dépendent que du dernier coup joué. Notons alors **Inn** la catégorie des arènes et stratégies innocentes. Pour montrer que **Inn** est cartésienne fermée, on se contente de remarquer qu'elle hérite sa structure cartésienne fermée de la catégorie des arènes et stratégies filaires. En effet, on peut vérifier que tous les morphismes fondamentaux relatifs à la structure symétrique monoïdale fermée de **Str** sont innocents (essentiellement, toutes les stratégies *copycat* sont innocentes). De même, toutes les stratégies innocentes sont clairement filaires puisque la P -vue courante est toujours incluse dans le fil courant, ainsi ce sont des morphismes de comonoïdes, et la catégorie **Inn** est cartésienne fermée. On a donc le théorème suivant :

Théorème 2.2.11 (Hyland/Ong). *La catégorie **Inn** est cartésienne fermée.*

Ainsi, **Inn** forme un modèle du λ -calcul simplement typé. Mais cette fois, c'est un modèle très proche de la complétude : on a un résultat de *définissabilité*, c'est-à-dire que si σ est une stratégie innocente *finie* (i.e. l'ensemble $\lceil \sigma \rceil$ est fini), on peut récupérer un λ -terme M simplement typé dont l'interprétation est σ . Bien sûr, pour préciser cet énoncé, il faudrait détailler de quel langage on parle, de quels types, et de quelle fonction d'interprétation. De façon générale, on se rend compte que les stratégies peuvent être vues comme des *formes normales* sur les termes, plus précisément ils correspondent aux arbres de Böhm, ces expansions éventuellement infinies des λ -termes par β -réduction et η -expansion. Cet aspect des stratégies innocentes servira de pierre d'angle aux travaux présentés dans la Partie I, on ne le décrit donc pas plus avant ici.

2.2.5 Structure cartésienne fermée

Dans cette sous-section, on décrit brièvement la structure cartésienne fermée obtenue sur **Inn**. On va donner les définitions des différentes composantes et constructions afférentes à cette structure, mais sans donner les détails des preuves, qui sont directes et sans surprises.

Projections. Commençons par décrire les produits. Comme le lecteur s'en doute, si A et B sont des arènes, c'est l'arène $A \times B$ qui jouera le rôle du produit. On définit les deux projections comme des stratégies *copycat* :

$$\begin{aligned} \pi_1 &= \{s \in \mathcal{L}_{A_1 \times B \Rightarrow A_2} \mid \forall s' \sqsubseteq^P s, s'_{\uparrow A_1} = s'_{\uparrow A_2}\} \\ \pi_2 &= \{s \in \mathcal{L}_{A \times B_1 \Rightarrow B_2} \mid \forall s' \sqsubseteq^P s, s'_{\uparrow B_1} = s'_{\uparrow B_2}\} \end{aligned}$$

Le comportement de π_1 et π_2 est simple à décrire ; il se contente de recopier les coups de l'Opposant, de gauche à droite et inversement. Notons qu'on aurait pu aussi définir les projections comme données par l'image des parties des identités par les injections naturelles plongeant les parties de $A \Rightarrow A$ dans $A \times B \Rightarrow A$ et les parties de $B \Rightarrow B$ dans $A \times B \Rightarrow B$.

Paires. Considérons maintenant deux stratégies $\sigma : C \Rightarrow A$ et $\tau : C \Rightarrow B$, on veut définir leur *paire* $\langle \sigma, \tau \rangle : C \Rightarrow A \times B$. L'idée est que chaque *fil* sur $C \Rightarrow A \times B$ est (à réadressage près) soit un fil de $C \Rightarrow A$, soit un fil de $C \Rightarrow B$. Ainsi, on peut donner la définition suivante (en cachant les étapes de réadressage) :

$$s \in \langle \sigma, \tau \rangle \Leftrightarrow \forall s' \sqsubseteq^P s, [s'] \in \sigma \vee [s'] \in \tau$$

Ces paires se comportent comme requis, c'est-à-dire qu'elles satisfont les axiomes de la Définition 1.2.1.

Objet terminal. L'arène 1 est terminale sur **Inn**. Les projections terminales $!_A : A \Rightarrow 1$ sont données par les stratégies réduites à $\{\epsilon\}$.

Curryfication. Notons que les arènes $A \times B \Rightarrow C$ et $B \Rightarrow (A \Rightarrow C)$ génèrent le même graphe, elles ne diffèrent que par réadressage. Soit κ ce réadressage, toute partie de $\mathcal{L}_{A \times B \Rightarrow C}$ peut être transformée en partie de $\mathcal{L}_{B \Rightarrow (A \Rightarrow C)}$ par l'action de κ , et réciproquement. Ainsi, on pose $\Lambda(\sigma) = \kappa \cdot \sigma$.

Évaluation. On pose simplement $ev_{A,B} = \Lambda^{-1}(id_{A \Rightarrow B})$.

2.2.6 Coproduits faibles fonctoriels

Coproduits. Les sommes en sémantique de jeux ont un passif mouvementé. Tout d'abord, rappelons que la disjonction en logique, ou la somme dans les langages de programmation, s'interprète naturellement dans une catégorie dotée de *coproduits*, c'est-à-dire qu'elle devrait être *bicartésienne fermée*. On recherche donc le moyen de construire des coproduits dans une catégorie de jeux.

Polarité. Les jeux présentés ici sont toujours *négatifs*, c'est-à-dire qu'Opposant joue toujours en premier. On peut bien sûr avoir un cadre plus général où on n'a pas cette contrainte, voir par exemple la catégorie **Con** des jeux de Conway [Joy77]. Notons alors que si le produit s'exprime très bien dans un cadre négatif (Opposant choisit la composante à interroger), la somme s'exprime bien dans un cadre positif (Joueur, par son premier coup, choisit la composante dans laquelle il va jouer). En particulier, distinguons les deux sous-catégories de **Con** des jeux *positifs* \mathbf{Con}^+ , où Joueur commence toujours, et les jeux *négatifs* \mathbf{Con}^- , où Opposant commence toujours. On a alors la propriété surprenante que \mathbf{Con}^+ a tous les coproduits et que \mathbf{Con}^- a tous les produits, alors que **Con** n'a ni les uns ni les autres [Mel05]. Aussi, il y a peu d'espoirs de trouver une catégorie de jeux qui soit bicartésienne puisque toutes les catégories de jeux sont en relation étroite avec une certaine sous-catégorie des jeux de Conway.

Complétion libre. On a l'impression d'avoir sonné le glas de la représentation des sommes en sémantique de jeux, pourtant il existe plusieurs solutions. L'une d'elles est de construire une catégorie cartésienne fermée de jeux, puis d'obtenir librement une catégorie bicartésienne fermée par la construction de familles [AM98]. C'est une méthode propre et algébrique, mais non interne à la sémantique de jeux. Or, le genre de développement combinatoires complexes sur les arènes présenté dans cette thèse nécessite une représentation *interne* des sommes, comme une arène standard.

Coproduits faibles. L'autre solution est celle choisie par exemple dans [McC00]. On choisit d'avoir une construction interne, au prix du caractère *strict* des coproduits. L'idée est que pour représenter $A + B$, on commence par plonger A et B dans une catégorie positive par l'intermédiaire d'un *décalage* [Lau02, Tab08] : on rajoute un coup, et on inverse la polarité des coups internes à A et B . Là on prend le coproduit, puis on repasse dans la catégorie négative par l'intermédiaire d'un autre décalage. On perd le caractère strict car les coups rajoutés viennent avec leur propre dynamique : ainsi a-t-on en général, si $\sigma : A \rightarrow C$ et $\tau : B \rightarrow C$, plusieurs stratégies $h : A + B \rightarrow C$ candidates pour la co-paire. Il est toujours possible ensuite de récupérer un coproduit (presque) fort *via* un quotient extensionnel bien choisi. Ainsi dans [McC00], c'est le quotient observationnel de [HO00] qui joue ce rôle, conduisant ainsi à un modèle en *pleine adéquation* avec le langage de programmation FPC. Dans cette thèse on veut garder la connection étroite entre les stratégies et les termes, ainsi on se contentera de la forme faible de cette somme.

Foncteurs. Afin d'interpréter une logique ou un calcul dans une catégorie dotée de coproduits faibles, il faut tout de même s'assurer que ceux-ci définissent un foncteur ; c'est en général le minimum qu'on attend des constructeurs. Dans les préliminaires catégoriques, on a donné une définition d'une catégorie dotée de *coproduits faibles fonctoriels*, c'est-à-dire de coproduits vérifiant des conditions assez faibles, mais suffisantes pour assurer la functorialité des constructions. On va maintenant montrer que les arènes de la forme $A + B$ introduites au début de ce chapitre équipent **Inn** de *coproduits faibles fonctoriels*.

Injections. Si A et B sont des arènes, alors on définit les injections $in_1 : A \Rightarrow A + B$ et $in_2 : B \Rightarrow A + B$ par :

$$\begin{aligned} in_1 &= \{\epsilon\} \cup \{qLs \in \mathcal{L}_{A_1 \Rightarrow A_2+B} \mid \forall s' \sqsubseteq s, s'_{\uparrow A_1} = s'_{\uparrow A_2}\} \\ in_2 &= \{\epsilon\} \cup \{qLs \in \mathcal{L}_{B_1 \Rightarrow A+B_2} \mid \forall s' \sqsubseteq s, s'_{\uparrow B_1} = s'_{\uparrow B_2}\} \end{aligned}$$

Co-paire. Si $\sigma : A \Rightarrow C$ et $\tau : B \Rightarrow C$, on définit la *co-paire* $[\sigma|\tau]$ par :

$$\begin{aligned} [\sigma|\tau] &= \{\epsilon, qCq\} \cup \\ &\quad \{qCqLs \mid qCs \in \sigma\} \\ &\quad \{qCqRs \mid qCs \in \tau\} \end{aligned}$$

Dans cette notation, on prend quelques libertés avec les pointeurs. Formellement, les coups qui pointaient dans s vers L ou R doivent dans la partie droite pointer vers qC .

Proposition 2.2.12. *La co-paire et les injections vérifient les propriétés suivantes :*

$$\begin{aligned} in_1; [\sigma|\tau] &= \sigma \\ in_2; [\sigma|\tau] &= \tau \end{aligned}$$

Démonstration. Vérification directe. □

Notons qu'en général on n'a pas $[\sigma|\tau];\delta = [\sigma;\delta|\tau;\delta]$. En effet, le terme de droite est toujours strict : il va directement consulter son argument, alors que celui de gauche peut répondre directement à droite si δ n'était pas stricte.

Lemme 2.2.13. *Soient $\sigma : A \Rightarrow C$ et $\tau : B \Rightarrow C$, soit $\delta : C \Rightarrow D$ une stratégie stricte, c'est-à-dire qu'elle joue immédiatement sur C dès réception du coup initial sur D . Alors :*

$$[\sigma|\tau];\delta = [\sigma;\delta|\tau;\delta]$$

Démonstration. Soit $s \in [\sigma|\tau];\delta$, il existe un témoin $u \in [\sigma|\tau] \parallel \delta$. Puisque δ est stricte, et par définition de la co-paire, u est soit vide, soit de la forme $q_D q_C q L u'$ (resp. $q_D q_C q R u'$), avec $q_D q_C u'$ une interaction dans $\sigma \parallel \delta$ (resp. $\tau \parallel \delta$). En particulier, on en déduit que s est de la forme $q_D q L s'$ (resp. $q_D q R s'$) avec s' dans $\sigma; \delta$ (resp. $\tau; \delta$), donc $s \in [\sigma;\delta|\tau;\delta]$. Réciproquement, si $s \in [\sigma;\delta|\tau;\delta]$, s est de la forme $q_D q L s'$ (resp. $q_D q R s'$) avec $q_D s'$ dans $\sigma; \delta$ (resp. $\tau; \delta$). Soit $q_D u'$ l'interaction témoin de $q_D s'$, $q_D u'$ est de la forme $q_D q_C u''$ puisque δ est stricte. Alors, une vérification directe montre que $q_D q_C q L u''$ (resp. $q_D q_C q R u''$) est bien un témoin pour s dans $[\sigma|\tau];\delta$. \square

Objet faiblement initial. L'arène 0 est faiblement initiale dans **Inn**. Les morphismes canoniques $\iota_A : 0 \rightarrow A$ sont les stratégies de la forme $\{\epsilon, q\circ\}$, c'est à dire qu'elles répondent à tout coup initial de A par l'unique coup de 0 .

De tous ces ingrédients, on déduit la proposition suivante :

Proposition 2.2.14. *La catégorie **Inn** a les coproduits faibles fonctoriels.*

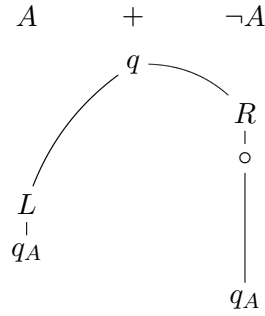
Démonstration. La vérification que $[in_1|in_2] = id$ est immédiate. Si $f : A \rightarrow C$, $g : B \rightarrow D$, $f' : C \rightarrow E$, $g' : D \rightarrow E$, l'égalité $[f; in_1|g; in_2]; [f'|g'] = [f; f'|g; g']$ découle directement de la Proposition 2.2.12 et du Lemme 2.2.13. \square

Remarque. Cette forme faible de la somme est suffisante pour interpréter bon nombre de calculs ; nous verrons par exemple qu'elle suffit pour interpréter la logique μLJ tout en gardant une forme de complétude. En pratique, on a toujours définissabilité des stratégies avec cette notion de somme à condition que dans la syntaxe, l'application ne commute pas avec le filtrage.

Remarque. La première version de la présente catégorie de jeux, introduite dans [H000], ne donnait pas d'interprétation pour les sommes. En fait une restriction sur la forme des arènes imposait qu'aucune réponse ne justifie de question. Cette contrainte laisse suffisamment de place pour définir l'arène des booléens de la Figure 2.1, mais interdit la définition de la somme donnée précédemment. Le lecteur pourra objecter avec raison qu'on ne s'est pas encore vraiment servi des questions/réponses, ce qui rend cette explication peu satisfaisante. En fait, la structure questions/réponses sert à assurer la *complétude* du modèle, notamment que toute stratégie $\sigma : A + B$ provient de $\sigma_A : A$ ou de $\sigma_B : B$.

2.3 Vers la complétude

Comme on l'a vu, la catégorie **Inn** est cartésienne fermée et a les coproduits faibles fonctoriels. Cependant, cette catégorie ne fournit pas encore de modèle complet d'un calcul intuitionniste. Notons $\neg A = A \Rightarrow 0$, on a alors une stratégie pouvant servir d'interprétation au tiers-exclus, dont les P -vues sont décrites par le diagramme suivant :



Ensuite, la stratégie copie d'un côté à l'autre les coups dans A . Puisqu'on veut dans cette thèse construire un modèle complet d'une logique intuitionniste (ou, question de point de vue, d'un langage de programmation fonctionnel sans opérateur de contrôle), il va falloir interdire ce genre de comportement. C'est la structure de questions/réponses qui va permettre de s'en débarrasser. L'idée est d'imposer un critère de *bon parenthésage* : chaque question devra recevoir sa réponse exactement une fois, lorsque toutes les questions qui la suivent auront reçu la leur. Par exemple, la stratégie ci-dessus sera interdite car la question initiale y reçoit deux fois une réponse.

2.3.1 Bon parenthésage

Définition 2.3.1. Soit A une arène, et $s \in \mathcal{L}_A$. Si s_j pointe vers s_i , que s_i est une question et s_j une réponse, on dit que s_j répond à s_i . Une question est dite *pendante* dans s si c'est la dernière question de s sans réponse. s est dite *bien parenthésée* si toute réponse pointe toujours sur la question pendante.

Remarque. La condition de parenthésage tient son nom de l'analogie avec les parenthèses. Ainsi, une question peut être vue comme une parenthèse ouvrante '(' et une réponse comme une parenthèse fermante ')'. La condition dit alors que le mot correspondant à la partie sur l'alphabet $\{(\,)\}$ est bien parenthésé.

Définition 2.3.2. Soit $\sigma : A$ une stratégie innocente. On dit que σ est bien parenthésée si pour tout $s \in \lceil \sigma \rceil$, s est bien parenthésée.

Montrons le lemme suivant, qui explique les interactions entre la P -vue et la condition de parenthésage.

Lemme 2.3.1. Soit A une arène et $s \in \mathcal{L}_A$ bien parenthésée. Si c'est à Joueur de jouer, alors la question pendante de s est la même que pour $\lceil s \rceil$.

Démonstration. La preuve se fait par induction sur $|s|$. Si le dernier coup est initial, alors c'est nécessairement une question de l'Opposant et l'énoncé est trivial ; ce cas comprend le cas de

base. Supposons que c'est à Joueur de jouer. Alors si m est initial m est nécessairement une question, pendante dans sm et aussi dans $\lceil sm \rceil = m$. Sinon, sm est de la forme s_1ns_2m où m pointe sur n , et $\lceil sm \rceil = \lceil s_1 \rceil nm$. Si m est une question, le résultat à montrer est trivial. Sinon c'est une réponse à n et par la condition de parenthésage la question pendante de sm est celle de s_1 . Mais l'hypothèse d'induction nous dit que c'est également la question pendante de $\lceil s_1 \rceil$, et donc de $\lceil s_1 \rceil nm = \lceil sm \rceil$ comme requis. \square

Lemme 2.3.2. *Soient $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies bien parenthésées et soit $u \in \sigma \parallel \tau$ une pré-vue, alors u est bien parenthésée.*

Démonstration. Montrons-le par induction sur u , en maintenant dans l'hypothèse d'induction que si c'est à σ (resp. τ) de jouer, alors la question pendante de u est dans $A \Rightarrow B$ (resp. $B \Rightarrow C$). Si u est vide c'est évident, sinon considérons une pré-vue ua . Si a est une question c'est évident, considérons donc le cas où a est une réponse. Si c'est une réponse de σ ou τ , considérons sans perte de généralité qu'il s'agit de σ . Puisque σ est bien parenthésée, a pointe sur la question pendante de $\lceil u \upharpoonright_{A \Rightarrow B} \rceil$, mais c'est aussi la question pendante de $u \upharpoonright_{A \Rightarrow C}$ par le Lemme 2.3.1. Or puisque c'était à σ de jouer la question pendante de u était bien celle de $u \upharpoonright_{A \Rightarrow C}$ et u est bien parenthésée. Si a est dans A il n'y a plus rien à montrer pour ce cas. Si a est dans B , alors ua peut être décomposée en u_1qu_2a de sorte que a répond à q . Mais alors q est nécessairement un coup de τ dans B , ce qui veut dire que par hypothèse d'induction, la question pendante de u_1 était dans $B \Rightarrow C$, or c'est également la question pendante de ua et c'est bien à τ de jouer. Reste le cas où a est une réponse de l'Opposant externe, mais celui-ci pointe nécessairement sur le coup précédent puisque u est une pré-vue, donc ce coup précédent était nécessairement une question pendante et ua est bien parenthésée. u était en fait de la forme $u'qa$ et q était un coup de σ (resp. τ). Mais la question pendante de $u'qa$ est celle de u' , or elle était dans $A \Rightarrow B$ (resp. $B \Rightarrow C$) par hypothèse d'induction sur u' . Or comme a est un coup sur $A \Rightarrow C$, c'est toujours à σ (resp. τ) de jouer, ce qui conclut la démonstration. \square

On déduit facilement de ce lemme la stabilité par composition des stratégies bien parenthésées :

Proposition 2.3.3. *Supposons $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ bien parenthésées. Alors $\sigma; \tau$ est toujours bien parenthésée.*

Démonstration. Considérons une pré-vue $u \in \sigma \parallel \tau$. Par le Lemme 2.3.2, u est bien parenthésée. Restreindre à A, C ne fait qu'enlever des paires question/réponse associées, donc $u \upharpoonright_{A, C}$ est toujours bien parenthésée. Or, toute P -vue $s \in \lceil \sigma; \tau \rceil$ a une telle pré-vue témoin, donc $\sigma; \tau$ est bien parenthésée. \square

Il est immédiat de vérifier que l'identité est une stratégie bien parenthésée, ainsi que toutes les stratégies fondamentales relatives à la structure cartésienne fermée (avec coproduits faibles fonctoriels) de **Inn**. On obtient donc une catégorie **Par** cartésienne fermée et équipée de coproduits faibles fonctoriels, dont les objets sont les arènes et les morphismes sont les stratégies innocentes bien parenthésées. C'est cette catégorie et ses variantes qui sont à la base de nombreux modèles complets de logiques et langages de programmation.

2.3.2 Définissabilité

Dans cette sous-section, on va décrire la procédure de *définissabilité*, permettant de recomposer inductivement un terme à partir d'une stratégie innocente et bien parenthésée. Puisqu'on

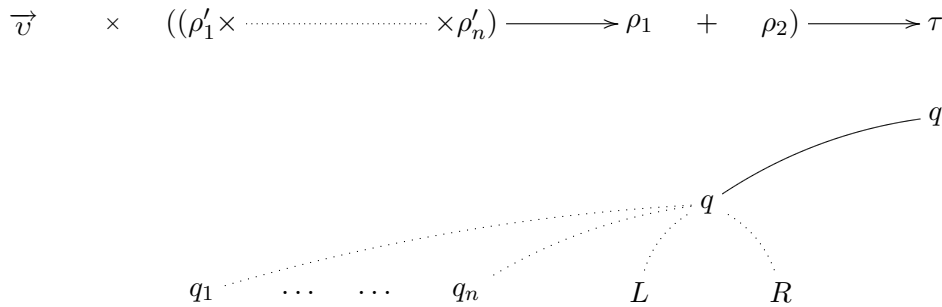
veut que cette partie préliminaire reste modulaire et pas associée à une logique ou un langage de programmation en particulier, on ne fixera pas de syntaxe : à la place, on utilisera comme syntaxe le langage des catégories cartésiennes fermées avec coproduits faibles fonctoriels. La procédure de définissabilité se fait par induction sur les types, aussi on se donne une syntaxe de *types simples* :

$$\tau ::= 0 \mid 1 \mid \tau + \tau \mid \tau \times \tau \mid \tau \rightarrow \tau$$

À chacun de ces types simples on associe une *interprétation* $\llbracket \tau \rrbracket$ par induction sur τ , chaque constructeur étant interprété par le constructeur correspondant de la structure catégorique de **Par**. Remarquons également que **Par** est *pointée*, c'est-à-dire que chaque arène A est habitée par une stratégie $\perp : A$ ne répondant pas à l'éventuel coup initial de l'Opposant. Supposons alors qu'on ait $\sigma : \llbracket \tau \rrbracket$ innocente et bien parenthésée, raisonnons par induction sur τ :

Cas périphériques. Il se produit de nombreux cas périphériques où la procédure de définissabilité est simplifiée. On ne va pas les traiter tous ici, le lecteur est invité à consulter [McC96] pour plus de détails. Décrivons tout de même informellement ces cas : si σ ne répond pas, alors elle est donnée par la stratégie \perp . Si τ a plusieurs coups initiaux, on se ramène par isomorphismes de types au cas où c'est un produit, et l'innocence de σ permet de se ramener à la procédure de définissabilité sur chaque composante du produit. Si c'est une somme, l'innocence et le bon parenthésage montrent que σ doit être de la forme $\sigma'; in_1$ ou $\sigma'; in_2$, et on continue la procédure sur σ' .

Cas principal. À l'aide de raisonnements similaires et par isomorphismes de types, on se ramène au cas principal où $\sigma : \llbracket \vec{v} \times \rho \rightarrow \tau \rrbracket$, où ρ lui-même a la forme : $\rho = \rho'_1 \times \dots \times \rho'_n \rightarrow \rho_1 + \rho_2$, et où σ joue directement sur le coup initial de ρ . Le comportement connu de σ est alors représenté par le diagramme suivant, avec les réactions possibles de l'Opposant dans une P -vue de σ .



Toute P -vue de σ commence par l'un de ces préfixes. On va alors extraire $n + 2$ sous-stratégies de σ , une pour chacun de ces préfixes. On pose les définit par leurs P -vues :

$$\begin{aligned} \lceil \mathbf{arg}_1(\sigma) \rceil &= \{q_1 s \mid qq_1 s \in \lceil \sigma \rceil\} & : & \llbracket \rho'_1 \rrbracket \\ & \dots & & \\ \lceil \mathbf{arg}_n(\sigma) \rceil &= \{q_n s \mid qq_n s \in \lceil \sigma \rceil\} & : & \llbracket \rho'_n \rrbracket \\ \lceil \mathbf{cont}_1(\sigma) \rceil &= \{s \mid qqLs \in \lceil \sigma \rceil\} & : & \llbracket \vec{v} \times (\rho'_1 \times \dots \times \rho'_n \rightarrow \rho_1 + \rho_2) \rightarrow \tau \rrbracket \\ \lceil \mathbf{cont}_2(\sigma) \rceil &= \{s \mid qqRs \in \lceil \sigma \rceil\} & : & \llbracket \vec{v} \times (\rho'_1 \times \dots \times \rho'_n \rightarrow \rho_1 + \rho_2) \rightarrow \tau \rrbracket \end{aligned}$$

Ces stratégies sont bien définies, leur innocence et leur bon parenthésage sont conséquences directes de celles de σ . Ce sont des stratégies dont les P -vues sont strictement plus courtes

que celles de σ , on peut donc par hypothèse d'induction les définir dans le langage de termes catégoriques, et σ peut être définie à son tour à partir de ces stratégies. Cette procédure permet bien de récupérer un terme à partir d'une stratégie finie. Pour plus de détails, le lecteur pourra consulter [McC96], où cette procédure est décrite en détails pour une catégorie de jeux très similaire à **Par**.

2.4 Préliminaires spécifiques

On a fini d'introduire la catégorie **Inn** des arènes et stratégies innocentes, avec toute sa structure de base. Ils nous reste cependant à présenter quelques préliminaires supplémentaires sans liens forts entre eux, mais qui serviront pour des parties spécifiques de la thèse. Tout d'abord, il nous faut présenter quelques éléments de la structure développée dans **Inn** par McCusker [McC00] pour donner son modèle pleinement abstrait de FPC, c'est-à-dire des types récursifs. Ensuite, nous présenterons le théorème de Laurent [Lau05] donnant une condition nécessaire et suffisante sur les arènes pour que les objets correspondants dans **Inn** soient isomorphes au sens catégorique.

2.4.1 Types récursifs à la McCusker

Pour préparer le terrain au modèle de jeux de l'induction et de la coinduction développé dans la troisième partie, il convient de présenter la manière usuelle de représenter des types infinis en sémantique de jeux, comme formalisée par McCusker. Construire un modèle des types récursifs revient à savoir résoudre de façon canonique des équations récursives d'arènes de la forme :

$$X = F(X, X)$$

et ce pour une classe de foncteurs $F : \mathbf{Inn}^{op} \times \mathbf{Inn} \rightarrow \mathbf{Inn}$ la plus large possible. On veut donc construire un *point fixe* de F . La méthode, usuelle, est celle du théorème du point fixe de Knaster-Tarski : il s'agit d'itérer le foncteur F à l'infini, et d'examiner les cas dans lesquels cette opération converge.

Définition 2.4.1. *Soient A et B deux arènes. On définit l'ordre \sqsubseteq sur les arènes par $A \sqsubseteq B$ si et seulement si :*

$$\begin{aligned} M_A &\subseteq M_B \\ \lambda_A &= \lambda_{B \upharpoonright M_A} \\ \vdash_A &= \vdash_B \cap (M_A \times M_A) \\ I_A &= I_B \cap M_A \end{aligned}$$

Lemme 2.4.1. *L'ordre \sqsubseteq définit un ordre partiel complet. Son plus petit élément est l'arène vide I et les bornes supérieures des ensembles dirigés sont donnés par l'union composante par composante.*

Le lemme ci-dessus permet de résoudre de nombreuses équations récursives d'arènes. En effet, soit F un bifoncteur sur **Inn** continu et monotone par rapport à \sqsubseteq , et considérons l'équation :

$$X = F(X, X)$$

On peut en construire une solution simplement. On définit :

$$\begin{aligned} D_0 &= I \\ D_{n+1} &= F(D_n, D_n) \end{aligned}$$

On pose ensuite $D = \bigsqcup_{n=0}^{\infty} D_n$, qui vérifie bien $F(D, D) = D$. Cette construction permet donc de donner des solutions aux équations récursives d'arènes, mais il convient de vérifier que le résultat obtenu est canonique. La notion utilisée pour exprimer cette canonicité est celle d'*invariant minimal* [Fre90, Fre91, Fre92], qui permet de capturer abstraitement les propriétés algébriques garanties par la coïncidence limites/colimites observée sur les objets définis récursivement dans les domaines de Scott [SP82]. McCusker montre [McC00] que la construction ci-dessus donne bien des invariants minimaux, du moins à condition que F satisfasse des conditions que nous allons introduire maintenant.

Définition 2.4.2. *Supposons que $A \trianglelefteq B$. On définit les injections et projections de cette paire par :*

$$\begin{aligned} in_{A \trianglelefteq B} &= id_{B \downarrow A \Rightarrow B} \\ pr_{A \trianglelefteq B} &= id_{B \downarrow B \Rightarrow A} \end{aligned}$$

Définition 2.4.3 (Foncteurs fermés). *Les endofoncteurs fermés au sens de McCusker sont ceux dont l'action peut être internalisée par une transformation naturelle. Formellement, on les définit en trois étapes :*

- Soit $F : \mathbf{Inn} \rightarrow \mathbf{Inn}$ un foncteur sur \mathbf{Inn} . F est fermé si et seulement si il existe une transformation naturelle en A et B :

$$f_{A,B}^F : A \Rightarrow B \rightarrow F(A) \Rightarrow F(B)$$

Telle que pour tout $\sigma : A \rightarrow B$, $'F(\sigma)' = '\sigma'; f_{A,B}^F$.¹¹

- Soit $F : \mathbf{Inn}^{op} \rightarrow \mathbf{Inn}$, F est fermé si et seulement si il existe une transformation naturelle en A et B :

$$f_{A,B}^F : B \Rightarrow A \rightarrow F(A) \Rightarrow F(B)$$

Telle que pour tout $\sigma : B \Rightarrow A$, $'F(\sigma)' = '\sigma'; f_{A,B}^F$.

- Soit $F : (\mathbf{Inn}^{op})^k \times \mathbf{Inn}^p \rightarrow \mathbf{Inn}$, F est fermé si et seulement si il est fermé composante par composante.

Théorème 2.4.2 (McCusker). *Soit F un endofoncteur continu pour \trianglelefteq , supposons qu'il préserve les injections et projections canoniques, et qu'il est fermé au sens de McCusker. Alors $D = \bigsqcup_{i=1}^{\infty} F^n(I)$ est un invariant minimal pour F .*

Ce résultat se généralise aux foncteurs de variance mixte, et d'arité quelconque. Le lecteur est invité à se référer à [McC00] pour plus de détails.

2.4.2 Isomorphismes en sémantique de jeux

Dans cette sous-section, on va présenter un outil puissant des jeux d'arènes, permettant de réduire les isomorphismes au sens catégorique dans \mathbf{Inn} à des isomorphismes plus concrets sur les arènes. Ce théorème, présenté d'abord par Laurent dans le cadre d'arènes arborescentes [Lau05], est généralisé ici aux arènes cycliques.

¹¹On rappelle que si $\sigma : A \rightarrow B$, $'\sigma' : 1 \rightarrow A \Rightarrow B$ est le nom de σ , que l'on peut définir par $'\sigma' = \Lambda(\pi_1; \sigma)$.

Notation. On note ip la fonction de *préfixe instantané*, définie par $ip(\epsilon) = \epsilon$ et $ip(sa) = s$. De même, on note jp la fonction de *préfixe justifieur*, c'est-à-dire que $jp(\epsilon) = \epsilon$, $jp(si) = \epsilon$ si i est initial, et $jp(s_1ns_2m) = s_1n$ si m pointe sur n .

Définition 2.4.4. Soit A une arène. Un chemin sur A est un chemin au sens de la théorie des graphes, c'est-à-dire que c'est une suite finie $x_1x_2\dots x_n$ telle que $x_1 \in I_A$, et pour tout $i \in \{1, \dots, n-1\}$ $x_i \vdash_A x_{i+1}$. On note \mathcal{C}_A l'ensemble des chemins sur A . Les chemins sont un cas particulier de parties légales, c'est-à-dire qu'on a toujours $\mathcal{C}_A \subseteq \mathcal{L}_A$.

Définition 2.4.5. Soient A et B deux arènes. Un morphisme de chemins de A à B est une application $\phi : \mathcal{C}_A \rightarrow \mathcal{C}_B$ qui commute avec le préfixe, c'est-à-dire que :

$$\phi \circ ip = ip \circ \phi$$

De même, un morphisme de partie de A à B est une application $\phi : \mathcal{L}_A \rightarrow \mathcal{L}_B$ qui commute avec les deux notions de préfixe, c'est-à-dire :

$$\begin{aligned} \phi \circ ip &= ip \circ \phi \\ \phi \circ jp &= jp \circ \phi \end{aligned}$$

En fait, les deux notions ci-dessus coïncident : un morphisme de chemins contient doré et déjà suffisamment d'information pour générer un morphisme de parties de façon unique.

Proposition 2.4.3. Soient A et B deux arènes, et $\phi : \mathcal{C}_A \rightarrow \mathcal{C}_B$, alors il existe un unique morphisme de parties $\phi^* : \mathcal{L}_A \rightarrow \mathcal{L}_B$ tel que si $\phi^*|_{\mathcal{C}_A} = \phi$.

Démonstration. Soit $s \in \mathcal{L}_A$. Montrons par induction sur s qu'il y a un seul choix possible de $\phi^*(s')$ pour tout $s' \sqsubseteq s$ satisfaisant les contraintes de morphisme de partie. Si $s = \epsilon$, on a nécessairement $\phi^*(s) = \phi(s) = \epsilon$. Sinon, la partie a la forme $sa \in \mathcal{L}_A$. Notons c_{sa} le *chemin courant* de sa , c'est-à-dire la sous-séquence de sa suivant les pointeurs de justification d'un coup initial jusqu'au dernier coup de sa . Par hypothèse d'induction, il y a un choix unique de $\phi^*(s')$ pour tout $s' \sqsubseteq s$ satisfaisant les contraintes de morphisme parties. Mais puisqu'on doit avoir $ip(\phi^*(sa)) = \phi^*(ip(sa)) = \phi^*(s)$, on sait que $\phi^*(sa)$ a la forme $\phi^*(s)b$, où b est un coup sur B . De même, le calcul $jp(\phi^*(s)b) = jp(\phi^*(sa)) = \phi^*(jp(sa))$ impose le choix du pointeur de b dans $\phi^*(s)$. Finalement, la commutation avec jp de ϕ^* nous donne que pour tout $s' \sqsubseteq sa$, on a par induction immédiate sur s' l'égalité $\phi(c_{s'}) = c_{\phi^*(s')}$. Ainsi, $c_{\phi^*(s)b} = \phi(c_{sa})$ ce qui force la valeur de b . On a bien montré l'existence et l'unicité du morphisme de parties ϕ^* étendant ϕ . \square

Pour cet énoncé, on a désigné par ϕ^* l'extension de ϕ en morphisme de parties pour clarifier la preuve. En pratique, la proposition ci-dessus permet d'assimiler les notions de morphisme de chemins et morphisme de parties, ainsi si ϕ est un morphisme de chemin, on désignera toujours par ϕ le morphisme de parties engendré.

Définition 2.4.6. Soit $s \in \mathcal{L}_{A \Rightarrow B}$. On dit que s est une partie en zig-zag si elle vérifie les trois conditions suivantes :

- Chaque coup Joueur suivant un coup Opposant dans A est dans B ;
- Chaque coup Joueur suivant un coup Opposant dans B est dans A ;
- $s \upharpoonright_A$ et $s \upharpoonright_B$ ont les mêmes pointeurs.

Si s ne vérifie que les deux premières conditions, alors s est pré-zig-zag. Dans ce cas, on désigne par \bar{s} l'unique partie pré-zig-zag dans $\mathcal{L}_{B \Rightarrow A}$ telle que $s \upharpoonright_A = \bar{s} \upharpoonright_A$ et $s \upharpoonright_B = \bar{s} \upharpoonright_B$ (à réadressage près).

Théorème 2.4.4 (Laurent). *Soient A et B deux arènes. On a un isomorphisme de chemins $\phi : \mathcal{C}_A \rightarrow \mathcal{C}_B$ si et seulement s'il existe un isomorphisme $i_\phi : A \rightarrow B$ dans **Inn**.*

Idée de preuve. La généralité supplémentaire donnée ici ne modifie pas la preuve, qui peut être consultée dans [Lau05]. Donnons ici quelques idées. Si $\phi : \mathcal{C}_A \rightarrow \mathcal{C}_B$ un isomorphisme de chemin. On définit :

$$\begin{aligned} i_\phi &= \{s \in \mathcal{L}_{A \Rightarrow B} \mid \forall s' \sqsubseteq^P s, \phi(s' \upharpoonright_A) = s' \upharpoonright_B\} \\ i_\phi^{-1} &= i_{\phi^{-1}} \end{aligned}$$

Ces stratégies sont clairement innocentes, puisqu'elles ne dépendent que du chemin courant le quel est toujours inclus dans la P -vue. Le fait qu'elles soient inverses l'une de l'autre est immédiat à partir du fait que ϕ et ϕ^{-1} soient inverses.

Réciproquement, si $i : A \Rightarrow B$ est un isomorphisme dans **Inn**, alors on montre d'abord par induction sur s que s est toujours une partie zig-zag, et que les parties s de i sont exactement les parties «à l'envers» \bar{s} de i^{-1} : il suffit pour ça de prendre $s \in id_A$ et de raisonner sur le témoin dans $i \parallel i^{-1}$ de s . Mais le fait que tout $s \in i$ soit une partie zig-zag revient à dire que i est généré par un morphisme de parties ϕ , qui lui-même vient d'un morphisme de chemins par la Proposition 2.4.3. \square

Un corollaire intéressant de ce théorème est que chaque arène (éventuellement cyclique) est équivalente à une arène arborescente, éventuellement infinie.

Définition 2.4.7 (Dépliage). *Soit A une arène. On appelle l'arène des parties ou le dépliage de A , noté \tilde{A} , l'arène suivante.*

$$\begin{aligned} M_{\tilde{A}} &= \mathcal{C}_A \setminus \{\epsilon\} \\ \lambda_{\tilde{A}} &= ca \mapsto \lambda_A(a) \\ \vdash_{\tilde{A}} &= \{(c, ca) \mid c, ca \in \mathcal{C}_A\} \\ I_{\tilde{A}} &= \{i \mid i \in I_A\} \end{aligned}$$

Les coups de \tilde{A} sont les chemins de A . Notons qu'il est inutile de leur spécifier des adresses, puisqu'on ne sera jamais amenés à effectuer une restriction sur une sous-arène de \tilde{A} .

Proposition 2.4.5. *Soit A une arène, alors A est isomorphe à sa forme dépliée. Notons $\gamma_A : A \rightarrow \tilde{A}$ l'isomorphisme.*

Démonstration. Par le Théorème 2.4.4, il suffit de donner un isomorphisme de chemins. Mais les chemins dans \tilde{A} sont des séquences $C = c_0 c_1 c_2 \dots c_n$, où chaque c_i est un chemin dans A et $c_i = ip(c_{i+1})$: on pose alors $\phi(C) = c_n$. On vérifie immédiatement que ϕ ainsi défini est un morphisme de chemins, et par ailleurs qu'il s'agit d'une bijection. \square

Remarque. Ce résultat permet d'expliquer pourquoi l'extension des arènes en arènes cycliques n'influe en aucune manière sur la construction de la catégorie **Inn**. C'est simplement parce qu'à isomorphisme près, la structure de l'arène n'est pas importante : seule compte la forêt des chemins générée.

Remarque. On a omis ici de parler des Questions/Réponses. La seule chose à garantir est que les isomorphismes γ_A et γ_A^{-1} soient bien parenthésés, et pour ça il suffit de demander également aux morphismes de chemins/ de parties de préserver la nature (Question/Réponse) des coups.

Première partie

Interactions et Totalité

Chapitre 3

λ -calcul et normalisation

Sommaire

3.1	λ-calcul unaire, définition et normalisation	68
3.2	Réduction linéaire de tête	72
3.2.1	Définitions	73
3.2.2	Propriétés	76
3.2.3	Normalisation	77
3.3	Machines abstraites	79
3.3.1	Réduction dirigée	79
3.3.2	Machine de Krivine	80
3.3.3	Machine à pointeurs	82

Dans ce chapitre, nous allons revisiter différents aspects de la connexion connue entre la réduction linéaire de tête du λ -calcul et l'interaction en sémantique de jeux [DHR96a]. En soi, ce chapitre ne contient pas de travaux particulièrement novateurs, mais est à prendre comme une introduction à cette relation, qui va être utilisée pour un premier examen (syntaxique) de la totalité en sémantique de jeux et inspirera ensuite une preuve sémantique, par réalisabilité, de préservation de la totalité par composition.

De façon surprenante on s'aperçoit que pour examiner ces questions de normalisation, il est inutile de considérer le λ -calcul en entier : il suffit de se restreindre aux termes *unaires*, en ce sens que les abstractions et les applications ne peuvent se faire que une à la fois. L'autre intérêt est que cette restriction simplifiera considérablement l'expression de la machine à pointeurs [DR]. Les types seront également plus simples : il est inutile de considérer tous les types simples pour typer ce calcul unaire, mais uniquement les types *purs*. En retour de ces simplifications, il nous faut rajouter à notre calcul un opérateur de *somme non déterministe*, ainsi on obtient un langage qui sera en correspondance directe avec les *structures de pointeurs* du Chapitre 4.

Dans la Section 3.1, nous définirons le λ -calcul unaire et verrons comment prouver sa normalisation de tête à l'aide de la réalisabilité intuitionniste. On démontrera aussi au passage la normalisation forte. Dans la Section 3.2, on introduira la réduction linéaire de tête pour ce calcul, et on déduira sa normalisation de celle de la réduction de tête. Ensuite, on verra dans la Section 3.3.3 les définitions des machines à pointeur et de Krivine sur ce calcul.

3.1 λ -calcul unaire, définition et normalisation

Comme annoncé, on va s'intéresser à des λ -termes *unaires*, qui ne peuvent prendre qu'un seul argument à la fois. L'intérêt de ce langage est qu'il est notablement plus simple que le λ -calcul complet, notamment au niveau des types. Cependant, il reste complet pour la divergence, c'est-à-dire que toute éventuelle chaîne infinie de réduction pour un λ -terme simplement typé peut être transformée en une chaîne infinie de réduction pour un λ -terme unaire, du moins avec l'ajout d'un opérateur de choix non déterministe. On définit les λ -termes unaires comme suit :

Termes. On considère un ensemble dénombrable de *variables* $x, y, z \dots$. Les termes sont obtenus par la grammaire suivante :

$$M, N ::= (M)N \mid \lambda x.M \mid M \mid M + N \mid \boxtimes$$

On appelle *termes* les mots générés par la grammaire ci-dessus.

Types. Les types considérés sont juste des types *purs*, c'est-à-dire qu'ils sont composés uniquement d'un atome de base o , et qu'ils sont entièrement parenthésés à gauche. Autrement dit, chacun d'entre eux est uniquement déterminé par un entier $n \in \mathbb{N}$, par la bijection suivante :

$$\begin{aligned} \underline{0} &= o \\ \underline{n+1} &= \underline{n} \rightarrow o \end{aligned}$$

À cause de cette correspondance, on notera un type t uniquement par \underline{k} tel que $\underline{k} = t$. Voici les règles de typage :

λ -calcul unaire	
$\frac{}{\Gamma, x : \underline{k} \vdash x : \underline{k}} ax$	$\frac{}{\Gamma \vdash \boxtimes : \underline{k}} dem$
$\frac{\Gamma, x : \underline{k} \vdash M : \underline{0}}{\Gamma \vdash \lambda x.M : \underline{k+1}} lam$	$\frac{\Gamma \vdash M : \underline{k+1} \quad \Gamma \vdash N : \underline{k}}{\Gamma \vdash MN : \underline{0}} app$
$\frac{\Gamma \vdash M : \underline{k} \quad \Gamma \vdash N : \underline{k}}{\Gamma \vdash M + N : \underline{k}} plus$	

On équipe ce calcul des règles de réduction non déterministes suivantes :

$$\begin{aligned} (\lambda x.M)N &\rightsquigarrow M[N/x] && (\beta) \\ \boxtimes N &\rightsquigarrow \boxtimes && (\boxtimes) \\ M + N &\rightsquigarrow M && (+_l) \\ M + N &\rightsquigarrow N && (+_r) \end{aligned}$$

Préservation des réductions. On a prétendu plus haut que toute éventuelle chaîne de réduction infinie pour un λ -terme simplement typé donnait lieu à une chaîne de réduction infinie pour un λ -terme unaire. Sans rentrer dans les détails, donnons-en une brève justification. On ne donnera pas ici la définition du λ -calcul simplement typé, le lecteur pourra se référer à [Bar85]. On définit une traduction $(\)^*$ sur les termes du λ -calcul simplement typé :

$$\begin{aligned} (\lambda x_1 \dots x_n. M)^* &= \lambda x. (M[x/x_i])^* \\ (M U_1 \dots U_p)^* &= M^* (U_1^* + \dots + U_p^*) \\ x^* &= x \end{aligned}$$

On peut vérifier que la traduction d'un terme simplement typé est toujours un terme typé du λ -calcul unaire, et que cette traduction transporte les chaînes de réduction du λ -calcul simplement typé vers le λ -calcul unaire. On ne va cependant pas en donner les détails, car cette traduction ne sera pas utile techniquement pour la suite de cette thèse. Elle sous-tend cependant l'écrasement qui sera présenté dans le cadre des jeux en Section 4.1.1, et permet d'expliquer la présence de l'opérateur de somme non déterministe dans le λ -calcul unaire.

Normalisation. Pour le moment, on s'intéressera uniquement à la réduction de tête, *i.e.* on réduit uniquement l'éventuel redex le plus à gauche. On notera Λ l'ensemble des *termes* de ce calcul. De même, on désignera par Λ^0 l'ensemble des *termes clos*. Notons qu'un terme non typé dans Λ^0 n'est pas forcément unaire, puisque c'est le typage qui garantit cette condition. Dans la suite, on précisera explicitement à chaque fois qu'on considérera un terme typé.

Définition 3.1.1 (Réalisation). *On définit une relation $\Vdash \subseteq \Lambda^0 \times \mathbb{N}$:*

- $M \Vdash 0 \Leftrightarrow M \rightsquigarrow^* \boxtimes$
- $M \Vdash k + 1 \Leftrightarrow \forall N \Vdash k, MN \rightsquigarrow^* \boxtimes$.

Lemme 3.1.1 (Adéquation). *Supposons $x_1 : \underline{n_1}, \dots, x_p : \underline{n_p} \vdash M : \underline{k}$. Alors, pour tout $N_1 \Vdash n_1, \dots, N_p \Vdash n_p$, $M[N_1/x_1, \dots, N_p/x_p] \Vdash k$.*

Démonstration. Par induction sur la dérivation de typage de M . On distingue les cas selon la dernière règle utilisée.

- *ax, dem.* Trivial.
- *lam.* La dernière règle est :

$$\frac{\Gamma, x : \underline{k} \vdash M : \underline{0}}{\Gamma \vdash \lambda x. M : \underline{k+1}} \textit{lam}$$

Soient $N_1 \Vdash n_1 \dots N_p \Vdash n_p$. Il faut montrer que $(\lambda x. M)[N_i/x_i] \Vdash k + 1$. Supposons $N \Vdash k$, alors :

$$((\lambda x. M)[N_i/x_i])N = (\lambda x. (M[N_i/x_i]))N \rightsquigarrow M[N_i/x_i][N/x] \rightsquigarrow^* \boxtimes$$

La première réduction est celle correspondant à l'unique redex de tête disponible. Le terme $M[N_i/x_i][N/x]$ normalise ensuite par hypothèse d'induction sur M .

- *app.* La dernière règle est :

$$\frac{\Gamma \vdash M : \underline{k+1} \quad \Gamma \vdash N : \underline{k}}{\Gamma \vdash MN : \underline{0}} \textit{app}$$

Soient $N_1 \Vdash n_1 \dots N_p \Vdash n_p$. Il faut montrer que $(MN)[N_i/x_i] \rightsquigarrow^* \boxtimes$. C'est juste par définition de la relation de réalisation, puisque par hypothèse d'induction, $M[N_i/x_i] \Vdash k + 1$ et $N[N_i/x_i] \Vdash k$.

– *plus*. La dernière règle est :

$$\frac{\Gamma \vdash M : \underline{k} \quad \Gamma \vdash N : \underline{k}}{\Gamma \vdash M + N : \underline{k}} \text{sum}$$

Soient $N_1 \Vdash n_1 \dots N_p \Vdash n_p$. Si $k = 0$, $(M + N)[N_i/x_i]$ se réduit en une étape sur $M[N_i/x_i]$ ou $N[N_i/x_i]$, or ces deux termes réalisent 0 par hypothèse d'induction, donc $(M + N)[N_i/x_i] \rightsquigarrow^* \mathbf{\star}$. Si $k \neq 0$, il nous faut montrer que $(M + N)[N_i/x_i] \Vdash k + 1$. Pour tout $T \Vdash k$, $((M + N)[N_i/x_i]T)$ se réduit en une étape, soit sur $(M[N_i/x_i]T)$ ou sur $(N[N_i/x_i]T)$, or ces deux termes convergent vers $\mathbf{\star}$ par hypothèse d'induction. \square

En réalisabilité, le lemme d'adéquation est un outil puissant, qui permet en particulier de montrer immédiatement la normalisation faible du calcul :

Corollaire 3.1.2. \rightsquigarrow normalise faiblement sur les termes typés, i.e. pour tout terme clos M tel qu'il existe $k \in \mathbb{N}$ avec $\vdash M : \underline{k}$ prouvable, la chaîne de réduction de M pour \rightsquigarrow est finie.

Démonstration. D'après le lemme d'adéquation, on a $M \Vdash k$. Soit $k = 0$ et $M \rightsquigarrow^* \mathbf{\star}$, soit $k = p + 1$. Dans ce cas, on note tout d'abord que $\mathbf{\star} \Vdash p$ (immédiat), donc on a $M\mathbf{\star} \rightsquigarrow^* \mathbf{\star}$. Mais si M avait une chaîne de réduction de tête infinie, celle-ci serait également une chaîne de réduction infinie pour $M\mathbf{\star}$, ce qui est impossible. \square

On notera la simplicité de cette démonstration. La thèse défendue dans ce chapitre est que le cœur du théorème de finitude des interactions en sémantique de jeux se trouve dans cette courte démonstration. La première preuve qu'on en donnera, en Section 4.2.1, consiste essentiellement en une traduction ramenant le théorème de finitude des interactions à la normalisation faible des λ -termes unaires. La preuve plus sémantique de la Section 4.2.2 consiste quant à elle plus ou moins à rejouer cette preuve par réalisabilité directement dans le langage des structures de pointeurs.

Techniquement, les développements présentés dans cette thèse n'utiliseront que le théorème de normalisation faible prouvé ci-dessus. Cependant, par souci de complétude, on montre brièvement comment l'étendre en un théorème de normalisation forte. Jusqu'à la fin de cette section, on considère désormais la relation de réduction \rightsquigarrow appliquée à n'importe quelle position des termes. On adapte le prédicat \Vdash de réalisabilité en un prédicat \Vdash_s plus fort :

Définition 3.1.2 (Realisabilité forte). On définit une relation $\Vdash_s \subseteq \Lambda^0 \times \mathbb{N}$:

- $M \Vdash_s 0$ si et seulement si toutes les chaînes de réduction de M convergent vers $\mathbf{\star}$;
- $M \Vdash_s k + 1 \Leftrightarrow \forall N \Vdash_s k, MN \Vdash_s 0$.

Comme ci-dessus, le fait qu'un terme est un réalisateur fort de k permet d'assurer qu'il est fortement normalisant. Plus précisément, on montre le lemme suivant :

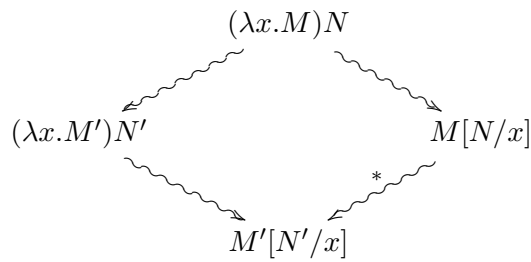
Lemme 3.1.3. Pour tout k , on a $\mathbf{\star} \Vdash_s k$. De plus, si $M \Vdash_s k$ alors M est fortement normalisant.

Démonstration. Par induction sur k . C'est clairement vrai pour $k = 0$. Montrons que $\mathbf{\star} \Vdash_s k + 1$. Pour ça, il faut considérer $M \Vdash_s k$ et montrer que $\mathbf{\star}M$ se normalise fortement sur $\mathbf{\star}$. Mais par hypothèse d'induction, on sait que M est fortement normalisant. Donc, toute chaîne de réduction finit par réduire le redex de tête, ce qui permet d'atteindre $\mathbf{\star}$. De la même façon,

montrons que si $M \Vdash_s k + 1$ alors M est fortement normalisant. Par hypothèse d'induction, on a $\mathfrak{X} \Vdash_s k$, donc $M\mathfrak{X}$ est fortement normalisant. Mais comme pour le cas de la normalisation faible, si M avait une chaîne de réduction infinie cette chaîne fonctionnerait aussi pour $M\mathfrak{X}$, donc M est fortement normalisant. \square

On montre l'adéquation de la même manière, mais le cas *lam* est plus subtil, il faut en effet se ramener au cas où on commence par réduire l'application en position de tête. On commence par prouver le lemme de permutation suivant :

Lemme 3.1.4. *Soient deux termes M et N . Supposons qu'on aie une réduction $(\lambda x.M)N \rightsquigarrow (\lambda x.M')N'$ qui ne soit pas une réduction de tête. Dans ce cas, on peut permuter cette réduction avec la réduction de tête, comme illustré :*

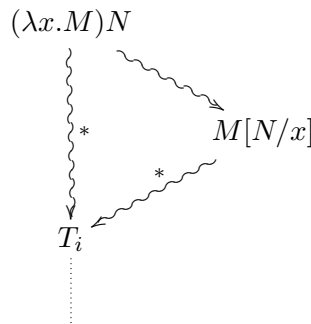


Démonstration. Si on fait dans le terme $(\lambda x.M)N$ une réduction qui n'est pas la réduction de tête, cette réduction doit être soit dans M , soit dans N . Supposons qu'elle soit dans M , on a alors $(\lambda x.M)N \rightsquigarrow (\lambda x.M')N$. Une induction immédiate sur la structure de M montre alors qu'on peut transporter cette réduction en $M[N/x] \rightsquigarrow M'[N/x]$, ce qui permet de fermer le diagramme.

Autrement, la réduction est dans N : on a $(\lambda x.M)N \rightsquigarrow (\lambda x.M)N'$. De même, la réduction $N \rightsquigarrow N'$ se transporte dans $M[N/x]$, en autant d'étapes qu'il y avait d'occurrences de x dans M . \square

Par itération de ce lemme, on obtient le résultat suivant :

Lemme 3.1.5. *On considère une chaîne de réduction infinie $(\lambda x.M)N \rightsquigarrow T_1 \rightsquigarrow T_2 \rightsquigarrow \dots$, avec M et N fortement normalisants. Alors, cette chaîne se factorise comme indiqué ci-dessous.*



Démonstration. Comme M et N sont fortement normalisants, toute éventuelle chaîne de réduction infinie de $(\lambda x.M)N$ finit forcément par réduire le redex de tête, soit $T_{i-1} \rightsquigarrow T_i$ cette réduction. Par induction sur i , on fait remonter cette réduction le long de la chaîne $(\lambda x.M)N \rightsquigarrow T_1 \rightsquigarrow T_2 \rightsquigarrow \dots$ jusqu'à obtenir la factorisation annoncée. \square

On peut alors montrer le lemme d'adéquation :

Lemme 3.1.6 (Adéquation forte). *Supposons $x_1 : \underline{n}_1, \dots, x_p : \underline{n}_p \vdash M : \underline{k}$. Alors, pour tout $N_1 \Vdash n_1, \dots, N_p \Vdash n_p$, $M[N_1/x_1, \dots, N_p/x_p] \Vdash_s k$.*

Démonstration. Par induction sur la dérivation de type. Les seuls cas qui diffèrent du lemme d'adéquation simple sont *lam* et *plus*.

– *lam*. La dernière règle est :

$$\frac{\Gamma, x : \underline{k} \vdash M : \underline{0}}{\Gamma \vdash \lambda x.M : \underline{k+1}} \textit{lam}$$

Soient $N_1 \Vdash_s n_1 \dots N_p \Vdash_s n_p$. Il faut montrer que $(\lambda x.M)[N_i/x_i] \Vdash_s k+1$. Supposons $N \Vdash_s k$. Commençons par noter que $M[N_i/x_i]$ et N sont fortement normalisants. En effet, N est fortement normalisant puisque $N \Vdash_s k$. De même, M est fortement normalisant, s'il avait une chaîne de réduction infinie, on pourrait la transporter dans $M[\mathfrak{X}/x_1, \dots, \mathfrak{X}/x_p, \mathfrak{X}/x]$ qui réalise 0 par hypothèse d'induction, ce qui est absurde. On doit montrer que $(\lambda x.M)[N_i/x_i]N = (\lambda x.M[N_i/x_i])N$ normalise fortement. Mais comme $M[N_i/x_i]$ et N sont fortement normalisants, on en déduit par le lemme précédent que toute réduction infinie de $(\lambda x.M)[N_i/x_i]N$ se factorise par la réduction de tête. Or, aucune réduction infinie ne peut commencer par la réduction de tête par hypothèse d'induction sur M .

– *plus*. La dernière règle est :

$$\frac{\Gamma \vdash M : \underline{k} \quad \Gamma \vdash N : \underline{k}}{\Gamma \vdash M + N : \underline{k}} \textit{sum}$$

On remarque que pour tout terme T , une chaîne de réduction de $(M+N)T$ qui ne réduit pas le redex de tête consiste en un entrelacement de chaînes de réduction de M , N et T . Soient $N_i \Vdash_s n_i$. Alors, il faut montrer que $(M+N)[N_i/x_i] = (M[N_i/x_i] + N[N_i/x_i]) \Vdash_s k$. Si $k = 0$, on commence par noter que toute éventuelle chaîne de réduction infinie de $(M[N_i/x_i] + N[N_i/x_i])$ finit tôt ou tard par réduire le redex de tête, car par hypothèse d'induction $M[N_i/x_i]$ et $N[N_i/x_i]$ sont fortement normalisants. Dès lors, on tombe sur un réduit de $M[N_i/x_i]$ ou de $N[N_i/x_i]$, qui sont fortement normalisants par hypothèse d'induction. Si $k = p+1$, on considère $T \Vdash_s p$. T étant fortement normalisant, le même raisonnement s'applique et $(M[N_i/x_i] + N[N_i/x_i])T$ se réduit forcément en un réduit de $M[N_i/x_i]T$ ou $N[N_i/x_i]T$ qui se réduisent fortement sur \mathfrak{X} , par hypothèse d'induction. \square

Ceci conclut la preuve de l'adéquation, et par le Lemme 3.1.3 on en déduit également la forte normalisation des termes bien typés. On précise encore une fois que si la forte normalisation de ce calcul n'est certes pas trop dure à obtenir, elle n'en reste pas moins d'un intérêt assez limité pour nous. En effet, tout ce travail sert de préliminaire à l'étude de la connection entre le calcul et les jeux, et dans les jeux la seule réduction qui nous intéresse est la réduction de tête. En fait, c'est même la réduction *linéaire de tête*, dont l'étude vient maintenant.

3.2 Réduction linéaire de tête

La *réduction linéaire de tête*, une réduction sur les λ -termes extrêmement paresseuse, semble être apparue pour la première fois sous ce nom dans [MP94], où elle est utilisée pour

montrer un théorème de séparation à la Böhm sur les réseaux de preuves. Cependant, on peut en trouver de nombreux précurseurs : citons notamment la β -réduction locale [DB87] étudiée par De Bruijn. Dans un travail de Danos, Herbelin et Regnier [DHR96a], elle est présentée comme le point de recoupement entre les jeux HO et AJM d’une part, et l’exécution de machines abstraites (la PAM¹ et la IAM² — la PAM sera décrite en détail plus tard, dans le cas des termes unaires). Dans un article suivant [DR] (à la connaissance de l’auteur, toujours non publié), Danos et Regnier reviennent plus en détail sur la réduction linéaire de tête et sur les relations qu’elle entretient avec les machines abstraites.

La réduction linéaire de tête joue un rôle fondamental dans cette partie. Elle sera aussi indispensable pour la première preuve de notre théorème de finitude, présentée dans la Section 4.2.1. Mais de façon plus générale, toute cette partie tire son inspiration de la relation entre la sémantique de jeux et la réduction linéaire de tête, et ce même lorsque nous aurons complètement laissé la syntaxe de côté et que nous parlerons uniquement d’objets sémantiques. La raison en est que les traces d’interaction de la sémantique de jeux sont des objets très difficiles à cerner et à contraindre, et finalement assez durs d’atteinte pour l’intuition. Il s’est avéré assez difficile de prouver des propriétés — même simples — sur ces objets, comme le lecteur pourra s’en aviser en Section 4.2. L’intuition de la réduction linéaire de tête, une fois transposée dans les traces d’interaction, permet d’avoir une approche concrète de ces objets abstraits, et de savoir mieux anticiper quelles propriétés on peut en attendre.

La réduction linéaire de tête *n’est pas* une stratégie de réduction au sens propre, d’une part parce que ses redex ne sont pas toujours des β -redex³, d’autre part parce que la substitution est *linéaire* : si on veut réduire $(\lambda x.M)N$, on ne va pas remplacer toutes les occurrences de x par M , mais *une seule*. De même, le sous-terme en position d’argument n’est pas effacé par la réduction, ce qui la rend strictement croissante en taille. Le lecteur familier avec le λ -calcul différentiel [ER03] notera que la réduction linéaire de tête partage avec ce calcul l’emploi crucial de la substitution linéaire.

On peut trouver dans [DR] une introduction à la réduction linéaire de tête dans le λ -calcul usuel. Dans cette thèse, on s’inspirera de cette présentation qu’on adaptera au λ -calcul avec l’opérateur de somme non déterministe introduit dans la section précédente.

3.2.1 Définitions

Le processus de réduction linéaire de tête peut être décrit en plusieurs étapes. Pour cette première description informelle, commençons par ignorer la somme non déterministe. Supposons donné un λ -terme T .

1. On repère l’*occurrence de tête* x_i de T : c’est l’occurrence de variable la plus à gauche dans l’écriture de T . En sémantique de jeux, c’est elle qui détermine le prochain coup joué par la stratégie.
2. On repère le *sous-terme d’abstraction* M correspondant à x_i . C’est le sous-arbre de T qui commence par $\lambda x. \dots$, où λx est l’abstraction qui lie x_i .
3. On repère le *sous-terme argument* N de M . C’est, s’il existe, le sous-terme qui est en position d’argument pour l’abstraction $\lambda x.$. Comme ce sera précisé plus tard, $(\lambda x, N)$ ne forme *pas nécessairement* un β -redex !
4. On effectue la substitution de x_i par N .

¹pour *Pointer Abstract Machine*, en français Machine Abstraite à Pointeurs.

²pour *Interaction Abstract Machine*, en français Machine Abstraite d’Interaction

³Bien que ce soit le cas pour le λ -calcul unaire

Habituellement, on n'a dans un terme T qu'une seule occurrence de tête, ce qui rend la réduction déterministe. Ici, la somme non déterministe permet d'avoir *plusieurs* occurrences de tête.

Notations et conventions. Dans cette section, on suivra les conventions de [DR]. Les lettres capitales seront utilisées pour désigner des termes, et les lettres minuscules x, y, z, \dots pour désigner des variables. Si x est une variable, on utilisera des indices x_0, x_1, x_2, \dots pour désigner les différentes *occurrences* de cette variable. De même, si T est un terme qui comprend plusieurs occurrences du démon (\boxtimes), on les désignera par $\boxtimes_1, \boxtimes_2, \dots$. On supposera aussi qu'on respecte la convention de Barendregt, c'est-à-dire que deux variables distinctes ont toujours un nom différent.

Définition 3.2.1. *Soit T un terme de Λ , on définit l'ensemble $hoc(T)$ de ses occurrences de tête par induction sur T , comme suit :*

$$\begin{aligned} hoc(T_1 T_2) &= hoc(T_1) \\ hoc(\lambda x.T) &= hoc(T) \\ hoc(x_i) &= \{x_i\} \\ hoc(T_1 + T_2) &= hoc(T_1) \cup hoc(T_2) \\ hoc(\boxtimes_j) &= \{\} \end{aligned}$$

L'idée, c'est qu'à chaque étape de réduction on va choisir (de façon non déterministe) une occurrence de tête. On va ensuite aller chercher le sous-terme argument correspondant, et effectuer la substitution. Pour associer formellement les variables abstraites aux sous-termes arguments qui leur correspondent, on utilise la définition de *pré-redex*. Plaçons-nous dans un terme T , un pré-redex de T est un couple $(\lambda x, M)$ avec x une variable abstraite dans T et M le sous-terme de T appliqué au sous-terme commençant par λx , c'est-à-dire celui qui est destiné à remplacer x . Bien sûr, tout cela sera défini formellement dans un instant. Dans le cadre actuel (*i.e.* non typé, avec la somme non déterministe), il peut être assez difficile de retrouver, étant donnée une abstraction, le sous-terme argument qui lui correspond. Cette difficulté explique la technicité de la définition qui suit. Bien sur, pour les termes unaires typés, tout ceci peut être drastiquement simplifié. Pour ce qui suit, on considèrera qu'une *forêt d'abstractions* est une structure de donnée définie par induction comme étant soit la forêt vide, soit un ensemble fini de couples $(\lambda x, F)$ où λx est une abstraction, et F est une forêt d'abstractions. Si F_1, \dots, F_n sont des forêts d'abstractions, on note $[F_1, \dots, F_n]$ la forêt obtenue en mettant les forêts F_1, \dots, F_n côte à côte, obtenant ainsi une nouvelle forêt.

Définition 3.2.2. *Soit T un terme de Λ , on commence par définir par induction la forêt des abstractions de T , notée $\lambda_h(T)$:*

- Si $T = T_1 T_2$, alors si $\lambda_h(T_1)$ est vide, $\lambda_h(T)$ est vide. Sinon, on a $\lambda_h(T_1) = \{(\lambda x^1, F_1), \dots, (\lambda x^n, F_n)\}$. Dans ce cas, on définit $\lambda_h(T) = [F_1, \dots, F_n]$.
- $\lambda_h(\lambda x.T') = \{(\lambda x, \lambda_h(T'))\}$
- $\lambda_h(x_i) = \emptyset$
- $\lambda_h(T_1 + T_2) = [\lambda_h(T_1), \lambda_h(T_2)]$
- $\lambda_h(\boxtimes_i) = \emptyset$

Maintenant, on peut définir l'ensemble noté $red(T)$ de ses pré-redex :

- Si $T = T_1T_2$ alors si $\lambda_h(T_1) = \emptyset$, $red(T) = red(T_1)$. Sinon, si $\lambda_h(T_1) = \{(\lambda x^1, F_1), \dots, (\lambda x^n, F_n)\}$, auquel cas on définit :

$$red(T) = red(T_1) \cup \{(\lambda x^1, T_2), \dots, (\lambda x^n, T_2)\}$$

- $red(\lambda x.T) = red(T)$
- $red(x_i) = \emptyset$
- $red(T_1 + T_2) = red(T_1) \cup red(T_2)$
- $red(\mathbf{x}_i) = \emptyset$

L'idée est d'associer à chaque abstraction du terme l'argument qui lui correspond. Dans le cas général, cet argument peut être non trivial à récupérer. Par exemple, considérons le terme $(\lambda x.\lambda y.M)N_1N_2$. Ici le pré-redex $(\lambda y, N_2)$ n'est pas un β -redex, et N_2 n'est sûrement pas en position d'argument pour le sous-terme $\lambda y.M$. On peut maintenant définir formellement la réduction linéaire de tête comme suit :

Définition 3.2.3. Soit M un terme. Si $x_i \in hoc(M)$, et si $(\lambda x, N) \in red(M)$ avec x_i occurrence de x , alors :

$$M \rightsquigarrow^{LHR} M[N/x_i]$$

Où $M[N/x_i]$ désigne M où seule l'occurrence x_i a été remplacée par N , en laissant le reste du terme inchangé. On passe sous silence les étapes nécessaires d' α -conversion nécessaires pour préserver la convention de Barendregt. Avant réduction, on appellera le triplet $(\lambda x, x_i, N)$ un redex pour la réduction linéaire de tête.

Exemples. Considérons d'abord un exemple sans somme. Dans ce cas, on ne peut avoir à chaque étape qu'une seule occurrence de tête, et la réduction est déterministe. Pour clarifier les choses, on va souligner l'occurrence de tête à chaque étape de réduction, et surligner les éléments du pré-redex correspondant.

$$\begin{aligned} (\overline{\lambda x}.\underline{xx})(\overline{\lambda y}.\underline{yy}) &\rightsquigarrow^{LHR} (\lambda x.(\overline{\lambda z}.\underline{zz})\overline{x})(\lambda y.\underline{yy}) \\ &\rightsquigarrow^{LHR} (\overline{\lambda x}.\underline{\lambda z}.\underline{xx})\overline{x}(\overline{\lambda y}.\underline{yy}) \\ &\rightsquigarrow^{LHR} (\lambda x.(\underline{\lambda z}.\overline{\lambda u}.\underline{uu})\overline{z})\overline{x}(\lambda y.\underline{yy}) \\ &\rightsquigarrow^{LHR} (\lambda x.(\overline{\lambda z}.\underline{\lambda u}.\underline{zu})\overline{z})\overline{x}(\lambda y.\underline{yy}) \\ &\rightsquigarrow^{LHR} (\overline{\lambda x}.\underline{\lambda z}.\underline{\lambda u}.\underline{xu})\overline{z}\overline{x}(\overline{\lambda y}.\underline{yy}) \\ &\rightsquigarrow^{LHR} (\lambda x.(\underline{\lambda z}.\underline{\lambda u}.\overline{\lambda v}.\underline{vv})\overline{u})\overline{z}\overline{x}(\lambda y.\underline{yy}) \\ &\rightsquigarrow^{LHR} \dots \end{aligned}$$

Cette chaîne de réduction s'étendant à l'infini. Considérons maintenant un exemple avec une somme non déterministe. On notera qu'on a plus forcément unicité de l'occurrence de tête. On gardera cependant les mêmes notations que ci-dessus, mais il y aura parfois un choix implicite de l'occurrence de tête à substituer. À chaque étape, on fera apparaître en couleur toutes les occurrences de tête.

$$\begin{aligned} (\overline{\lambda x}.\underline{x}(\lambda y.\underline{x + y}))(\overline{\lambda z}.\underline{z}) &\rightsquigarrow^{LHR} (\lambda x.(\overline{\lambda z}.\underline{z})(\overline{\lambda x}.\underline{x + y}))(\underline{\lambda z}.\underline{z}) \\ &\rightsquigarrow^{LHR} (\overline{\lambda x}.\underline{\lambda z}.\underline{\lambda u}.\underline{x + u})(\lambda y.\underline{x + y})(\overline{\lambda z}.\underline{z}) \\ &\rightsquigarrow^{LHR} (\lambda x.(\underline{\lambda z}.\underline{\lambda u}.\underline{\lambda v}.\underline{v} + \underline{u}))(\lambda y.\underline{x + y})(\underline{\lambda z}.\underline{z}) \end{aligned}$$

À ce point la réduction s'arrête, parce que ni v ni u ne font partie d'un pré-redex.

Sur la σ -équivalence. Comme on l'a dit, les pré-redex de la réduction linéaire de tête ne sont pas forcément des β -redex. Il paraît cependant important de préciser qu'ils le restent à σ -équivalence près. La σ -équivalence sur les λ -termes a été introduite par Regnier dans [Reg94], bien qu'elle ait été considérée auparavant sous le nom «équivalence par permutation», notamment chez de Vrijer [DV87]. Cette équivalence permet d'assimiler les λ -termes qui se traduisent dans les mêmes réseaux de preuve [Reg]. En particulier, si un terme se traduit dans un réseau avec n coupures, on veut pouvoir identifier n redex dans le terme originel. Techniquement, on définit la réduction suivante :

$$\begin{aligned} ((\lambda x.U)V)W &\rightarrow_{\sigma} (\lambda x.UW)V \\ (\lambda xy.U)V &\rightarrow_{\sigma} \lambda y.(\lambda x.U)V \end{aligned}$$

L'équivalence \cong_{σ} est définie comme la plus petite congruence générée par \rightarrow_{σ} . On peut alors noter que si les redex pour la réduction linéaire de tête ne sont pas nécessairement des β -redex, ils le restent à σ -équivalence près.

3.2.2 Propriétés

Correction vis-à-vis de la réduction de tête. Une propriété cruciale de la réduction linéaire de tête est qu'elle reste correcte aussi bien vis-à-vis de \rightsquigarrow que de β seule. Cela est exprimé par l'énoncé suivant :

Proposition 3.2.1. *Soit T un terme, supposons que $T \rightsquigarrow^{LHR} T'$, alors T et T' sont β -équivalents.*

Démonstration. Supposons que l'occurrence de tête substituée dans $T \rightsquigarrow^{LHR} T'$ soit z_i , et qu'elle soit substituée par un terme A_i . T a nécessairement la forme suivante :

$$T = \dots (\lambda z. \dots z_i \dots) A_i \dots$$

Dans ce cas, T' a la forme suivante :

$$T' = \dots (\lambda z. \dots A_i \dots) A_i \dots$$

On remarque donc que puisque la réduction linéaire de tête n'élimine pas le β -redex, il suffit de le réduire dans les deux cas pour obtenir le même terme. Écrivons T sous la forme :

$$T = C[(\lambda z.M)A_i]$$

et T' sous la forme :

$$T' = C[(\lambda z.M')A_i]$$

Où $C[\]$ est un *contexte*⁴. Alors, on a $T \rightsquigarrow^{\beta} C[M[A_i/z]]$ et $T' \rightsquigarrow C[M'[A_i/z]]$, mais on a aussi $M[A_i/z] = M'[A_i/z]$, puisqu'aucune occurrence de z ne peut apparaître librement dans A_i . \square

Non seulement \rightsquigarrow^{LHR} préserve la classe de β -équivalence, mais les réductions linéaires de tête d'un terme terminent si et seulement si ses \rightsquigarrow -réductions terminent également : c'est l'objet de la prochaine sous-section.

⁴Il ne paraît pas nécessaire de rappeler formellement la définition d'un contexte. Informellement, il s'agit d'un terme contenant des «trous» notés $[\]$, et écrire $C[M]$ revient à remplacer dans $C[\]$ tous les trous par M , sans effectuer d' α -conversion.

Séquence de substitutions. Pour exprimer la relation entre la réduction linéaire de tête et l'action des machines abstraites qu'on va étudier, on a besoin de garder trace des substitutions effectuées dans la chaîne de réduction. Si T est un terme, on définit ses *séquences de substitutions* comme suit.

Définition 3.2.4. Soit T_1 un terme, et $C = (T_1 \rightsquigarrow^{LHR} T_2 \rightsquigarrow^{LHR} T_3 \rightsquigarrow^{LHR} \dots)$ une séquence (éventuellement infinie) de réductions linéaires de tête. On définit la séquence de substitutions de C comme étant la chaîne

$$(z_1, A_1), (z_2, A_2), \dots$$

Où z_i est l'occurrence de tête de T_i substituée dans $T_i \rightsquigarrow^{LHR} T_{i+1}$ et A_i est le sous-terme de T_i par lequel z_i est substituée.

Si T est un terme, son arbre de substitutions est l'arbre des séquences de substitutions, pour toutes ses chaînes de réduction possibles.

3.2.3 Normalisation

Dans cette sous-section, on va montrer la normalisation «forte» de la réduction linéaire de tête sur les termes typés de Λ . Il est quelque peu maladroit de parler de normalisation forte dans ce cas précis, puisque la réduction reste une réduction de tête. En effet, le seul non déterminisme dans le choix des redex vient uniquement de la somme non déterministe, et nous allons montrer que ce non déterminisme n'influence pas la terminaison du calcul. Plus précisément, nous allons ramener ici la normalisation de la réduction linéaire de tête à celle de la réduction de tête usuelle. La preuve élémentaire que nous allons présenter est parue (restreinte au cas sans somme non déterministe) dans [CH09], mais ne semblait pas être connue auparavant : habituellement, les preuves reliant ces deux calculs le font par l'intermédiaire d'une machine abstraite, par exemple la KAM.

Sur la réduction de tête. Dans la preuve qui suit, on doit gérer un léger désaccord entre la façon dont les deux réductions gèrent la somme non déterministe. En effet, \rightsquigarrow sélectionne un des deux côtés :

$$M + N \rightsquigarrow M$$

De son côté, \rightsquigarrow^{LHR} va explorer parallèlement les deux composantes de chaque somme, et n'effacera jamais le $+$ en tête.

$$M_0 + N_0 \rightsquigarrow^{LHR} M_1 + N_0 \rightsquigarrow^{LHR} M_2 + N_0 \rightsquigarrow^{LHR} M_2 + N_1$$

C'est naturel, parce qu'on s'attend à ce que la réduction linéaire de tête augmente toujours la taille des termes, au contraire de ce qui arriverait si on sélectionnait un des côtés de la somme. Pour montrer la normalisation de \rightsquigarrow^{LHR} , il sera donc naturel de s'appuyer non pas sur la normalisation de tête de \rightsquigarrow , mais sur la normalisation de la β -réduction de tête (autrement dit, on ne considère plus les règles de \rightsquigarrow se rapportant au démon ou à la somme non déterministe), définie composante par composante sur les sommes. Plus précisément, on définit par induction

sur T l'ensemble $head(T)$ de ses sous-termes en position de tête :

$$\begin{aligned} head(T_1 T_2) &= \{T_1\} \cup head(T_2) \\ head(\lambda x.T) &= \{\lambda x.T\} \cup head(T) \\ head(x_i) &= \{x_i\} \\ head(T_1 + T_2) &= head(T_1) \cup head(T_2) \\ head(\mathbf{\boxtimes}_j) &= \{\mathbf{\boxtimes}_j\} \end{aligned}$$

La β -réduction de tête est alors la réduction qui ne réduit que les β -redex en position de tête, c'est à dire tels que $(\lambda x.M)N \in head(T)$. Sa normalisation peut être déduite facilement de la normalisation faible de \rightsquigarrow (par un argument à la König) mais de façon encore plus immédiate, elle se déduit du résultat de normalisation forte pour \rightsquigarrow . En effet, tout pas de réduction pour $\rightsquigarrow^{h,\beta}$ est en particulier un pas de réduction pour \rightsquigarrow .

Proposition 3.2.2. *Soit M un terme tel que $\Gamma \vdash M : \underline{k}$ est prouvable. Alors, toutes les séquences de réduction linéaire de tête de M sont finies.*

Démonstration. Considérons la syntaxe des termes temporairement étendue par un terme \boxed{T} pour chaque terme T , avec la règle associée :

$$\boxed{T} \rightsquigarrow^{pop} T$$

Cette règle s'applique à chaque fois qu'une boîte \boxed{T} arrive en position de tête. Supposons maintenant qu'on aie $\vdash M : \underline{k}$ avec une chaîne de réduction infinie

$$M \rightsquigarrow^{LHR} M_1 \rightsquigarrow^{LHR} M_2 \rightsquigarrow^{LHR} M_3 \rightsquigarrow^{LHR} \dots$$

On va raisonner par induction sur la taille de la chaîne de β -réduction de tête de M , qui est nécessairement finie puisque M est bien typé. On commence par décomposer la β -réduction en deux étapes :

1. *Substitution* ($\beta\Box$) $(\lambda x.M)N \rightsquigarrow_{\beta\Box} M[\boxed{N}/x]$
2. *Ouverture*. (\Box^{-1}) En une étape, on ouvre toutes les boîtes dans M .

Supposons que $M \rightsquigarrow^{\beta\Box} M'$. On peut alors transporter la chaîne de réduction infinie ci-dessus comme suit :

$$\begin{array}{ccccccc} M & \xrightarrow{LHR} & M_1 & \xrightarrow{LHR} & M_2 & \xrightarrow{LHR} & M_3 \xrightarrow{LHR} \dots \\ \downarrow \beta\Box & & & & & & \\ M' & \xrightarrow{LHR \vee pop} & M'_1 & \xrightarrow{LHR \vee pop} & M'_2 & \xrightarrow{LHR \vee pop} & M'_3 \xrightarrow{LHR \vee pop} \dots \end{array}$$

L'idée est qu'à chaque fois qu'on réduisait, dans la chaîne infinie supérieure, le même redex que pour $M \rightsquigarrow_{\beta\Box} M'$, on se contente dans la chaîne inférieure d'ouvrir la boîte en position de tête correspondante. Il est clair que la chaîne de réduction inférieure sera toujours infinie : en effet, chaque pas de réduction dans la chaîne supérieure correspond à un pas de réduction dans la chaîne inférieure.

Ensuite, il suffit d'enlever toutes les étapes d'ouvertures inutiles dans la chaîne de réduction inférieure.

$$\begin{array}{ccccccc}
 M' & \xrightarrow{LHR \setminus pop} & M'_1 & \xrightarrow{LHR \setminus pop} & M'_2 & \xrightarrow{LHR \setminus pop} & M'_3 \xrightarrow{LHR \setminus pop} \dots \\
 \downarrow \square^{-1} & & & & & & \\
 M'' & \xrightarrow{LHR} & M''_{\sigma(1)} & \xrightarrow{LHR} & M''_{\sigma(2)} & \xrightarrow{LHR} & M''_{\sigma(3)} \xrightarrow{LHR} \dots
 \end{array}$$

Ici, $\sigma : \mathbb{N} \rightarrow \mathbb{N}$ est une fonction strictement croissante, utilisée pour masquer les étapes disparues d'ouverture de boîte.

Le point clé est que si la séquence de réduction de M' est infinie, alors celle de M'' aussi. D'une part, c'est évident si on utilise qu'un nombre fini de fois la réduction *pop* dans la chaîne de M' . Dans le cas contraire, le nombre de réductions linéaires de tête doit être lui aussi infini — sinon, la séquence de réduction de M' se terminerait par une infinité de *pop*, ce qui est impossible puisqu'un *pop* ne crée *jamais* de *pop*-redex. Donc quoi qu'il arrive, il y a un nombre infini de réductions linéaires de tête dans la chaîne de M' , mais ce sont précisément les réductions qui sont préservées dans la chaîne de M'' ! Par conséquent, celle-ci doit être elle aussi infinie.

D'un autre côté, on sait par hypothèse d'induction que la chaîne de réduction linéaire de tête de M'' doit être finie, puisque $M \rightsquigarrow^{h\beta} M''$. Précisons que le cas de base est trivial, puisqu'une forme normale pour la β -réduction de tête est aussi une forme normale pour la réduction linéaire de tête. \square

3.3 Machines abstraites

Bien qu'il soit, comme on l'a vu, possible de définir directement la réduction linéaire de tête sur les termes et de raisonner dessus, on doit avouer qu'elle semble dénuée de l'aspect canonique de la β -réduction⁵, en partie puisque certains de ses redex ne sont pas des β -redex. Par contre, elle apparaît de façon beaucoup plus canonique comme la stratégie de réduction effectuée par nombre de machines abstraites, notamment la PAM, qui nous intéresse de par sa relation avec la sémantique de jeux, et la machine de Krivine (KAM). On va présenter ici ces deux machines abstraites, et montrer qu'elles implémentent la réduction linéaire de tête pour les termes de Λ .

3.3.1 Réduction dirigée

Comme nous allons le voir, la KAM et la PAM implémentent toutes les deux la réduction linéaire de tête. Quelques détails doivent cependant être précisés. Tout d'abord, comme on le verra à la prochaine sous-section, la KAM dans sa présentation habituelle effectue une réduction linéaire de tête *faible* : elle ne réduit pas sous les abstractions. Cela dit, c'est surtout une question de présentation et on y accordera pas plus d'attention que nécessaire.

Un autre écart entre les machines abstraites et la réduction linéaire de tête est leur gestion de la somme non déterministe. En fait, les machines abstraites simulent ce qu'on va appeler ici la réduction *dirigée* : quand on vient d'effectuer une substitution, on choisit la prochaine

⁵Mais c'est aussi à cause des choix précis de syntaxe pour le λ -calcul ; dans la notation postfixe de [DB87] la définition est beaucoup plus naturelle.

occurrence de tête uniquement dans le sous-terme qu'on vient de substituer, et plus dans l'ensemble du terme.

Définition 3.3.1. Soit $C = T_1 \rightsquigarrow^{LHR} T_2 \rightsquigarrow^{LHR} T_3 \rightsquigarrow^{LHR} \dots$ une chaîne de substitutions linéaires de tête. Considérons sa séquence de substitutions $(z_1, A_1), \dots$. C est dirigée si et seulement si pour tout $i \geq 1$, z_{i+1} est une occurrence de tête de A_i .

Si un terme n'a pas de somme non déterministe, son unique chaîne de réduction linéaire de tête est dirigée. Considérons l'exemple de réduction linéaire de tête avec somme donné en Section 3.2.1. La réduction y est aussi dirigée. Cependant, si à la prochaine étape on choisissait de substituer l'occurrence de tête u , la réduction cesserait d'être dirigée. En d'autres termes une réduction dirigée garde les sommes et augmente toujours la taille du terme, mais lorsqu'elle choisit un côté d'une somme non déterministe, ce choix est *définitif* : l'autre côté de la somme ne sera plus jamais visité.

3.3.2 Machine de Krivine

La machine de Krivine [Kri85] est donnée par un *état*, et par un ensemble de *transitions* sur ces états. A chaque étape du calcul, un état de la KAM va être donné par un triplet (M, σ, π) , où M est le terme en train de s'exécuter, σ est un *environnement* associant une fermeture (*i.e.* la donnée d'un terme et d'un environnement) à chaque variable libre de M , et π est la pile, *i.e.* la liste des termes en position applicative sur M . Quelques définitions préliminaires sont nécessaires avant de donner la définition de la machine de Krivine.

Définition 3.3.2. On définit par induction les fermetures et les environnements. Si M est un terme, $fv(M)$ désignera comme d'habitude l'ensemble des variables libres dans M .

- Une fermeture est un couple (M, σ) , noté M^σ , où M est un terme (non nécessairement clos) et σ est un environnement pour M . Notons \mathcal{C} l'ensemble des fermetures.
- Un environnement pour M est une application partielle $\sigma : fv(M) \rightarrow \mathcal{C}$ (vide si M est clos), et qui associe des fermetures aux variables libres.

Une fois qu'on a ces définitions, on définit un *état* pour la KAM comme étant un triplet (M, σ, π) , noté $M^\sigma \star \pi$, où M est un terme, σ est un environnement pour M , et π est une *pile*, c'est-à-dire une liste finie de fermetures se terminant par convention par le symbole π_0 de fin de pile. La pile π sur laquelle on a ajouté la fermeture c sera notée $c \cdot \pi$. On équipe ensuite la machine des transitions suivantes.

Transitions de la KAM

$$\begin{array}{lll}
 (M_1 M_2)^\sigma \star \pi & \rightarrow & M_1^\sigma \star M_2^\sigma \cdot \pi & (\text{push}) \\
 (\lambda x. M)^\sigma \star N^\tau \cdot \pi & \rightarrow & M^{\sigma \cup \{x \mapsto N^\tau\}} \star \pi & (\text{pop}) \\
 x_i^\sigma \star \pi & \rightarrow & \sigma(x_i) \star \pi & (\text{call}) \\
 (M_1 + M_2)^\sigma \star \pi & \rightarrow & M_1^\sigma \star \pi & (+_g) \\
 (M_1 + M_2)^\sigma \star \pi & \rightarrow & M_2^\sigma \star \pi & (+_d)
 \end{array}$$

La machine s'arrête lorsqu'aucune de ces transitions n'est définie. Plus précisément, cela arrive lorsqu'arrivent en tête un démon ou une variable sur laquelle l'environnement n'est pas défini, ou alors lorsque le terme en position de tête est une abstraction et que la pile est vide. Cette dernière condition d'arrêt fait que la machine de Krivine va effectuer la réduction

linéaire de tête *faible* : elle ne va pas réduire sous les abstractions. On peut bien sûr récupérer la réduction linéaire de tête complète en ajoutant la transition suivante :

$$(\lambda x.M)^\sigma \star \pi_0 \rightarrow M^\sigma \star \pi_0$$

Ainsi la réduction va se poursuivre. Si une occurrence de x arrive un jour en position de tête, celle-ci sera considérée comme une variable libre et le calcul s'arrêtera. Dans cette section, on se contentera des règles écrites plus haut : celles-ci constituent en effet la présentation habituelle de la KAM.

Exemple. Gardons l'exemple de $(\lambda x.xx)(\lambda y.yy)$. Chaque ligne est un des états successifs de la machine.

Terme	Environnement	Pile
$(\lambda x.xx)(\lambda y.yy)$	\emptyset	π_0
$\lambda x.xx$	\emptyset	$(\lambda y.yy, \emptyset) \cdot \pi_0$
xx	$\{x \mapsto (\lambda y.yy, \emptyset)\}$	π_0
x	$\{x \mapsto (\lambda y.yy, \emptyset)\}$	$(x, \{x \mapsto (\lambda y.yy, \emptyset)\}) \cdot \pi_0$
$\lambda y.yy$	\emptyset	$(x, \{x \mapsto (\lambda y.yy, \emptyset)\}) \cdot \pi_0$
yy	$\{y \mapsto (x, \{x \mapsto (\lambda y.yy, \emptyset)\})\}$	π_0

On s'arrête là, car l'écriture des environnements devient vite très verbeuse. Le lecteur averti notera que dans ces quelques lignes d'exécution de la KAM, on a en fait effectué *qu'un seul* équivalent de réduction linéaire de tête : c'est la quatrième ligne, lorsqu'on récupère dans l'environnement la valeur de x . Les autres étapes reviennent à cibler l'occurrence de tête que l'on doit substituer, tout en tenant l'environnement à jour.

Remarque. Si comme on l'a supposé on respecte la convention de Barendregt, la notation $\sigma \cup \{x \mapsto c\}$ ne devrait jamais causer d'ambiguïté puisque x ne peut pas être déjà définie dans σ . Dans le cas contraire, on peut fixer la convention qu'on écrase l'affectation courante de x dans σ , convention qui préserve la correction de la machine. Comme l'a montré Sylvain Lippi dans [Lip02], l' α -conversion n'est pas nécessaire.

On notera qu'un état de la KAM a la forme d'un arbre, avec un grand nombre de redondances. On peut donner une version plus économe en mémoire où l'état sera un graphe orienté acyclique : on aura des environnements partagés entre les différentes fermetures. Par souci de simplicité, on gardera la présentation en arbre dans cette thèse.

On va maintenant relier l'exécution de la machine de Krivine à la réduction linéaire de tête dirigée. Pour commencer, on va donner dans le cadre de la KAM une définition parallèle à la séquence de substitution de la réduction linéaire de tête.

Définition 3.3.3. Soit T_1 un terme, et soit $R = T_1^{\sigma_1} \star \pi_1, T_2^{\sigma_2} \star \pi_2, \dots$ une exécution de la KAM sur T . La séquence de substitution de R est la séquence $(z_1, A_1), (z_2, A_2), \dots$ où les z_i, A_i sont respectivement l'occurrence en position de tête et le terme par lequel z_i est remplacé lors des transitions (call) de R .

Comme pour le cas de la réduction linéaire de tête, l'arbre de substitution de T_1 est l'arbre de ses séquences de substitutions, pour toutes ses exécutions par la KAM possibles.

Théorème 3.3.1. Soit T un terme, alors l'arbre des substitutions dirigées et faibles (i.e. l'arbre des substitutions pour toutes les chaînes de réduction linéaire de tête faible dirigées, sans réduire sous les abstractions) et l'arbre des substitutions (au sens de la KAM) de T coïncident.

Idée de preuve. La preuve est donnée dans [DR] sans les sommes, donnons ici les contours d'une preuve différente. On définit les λ -termes *pointés* comme étant des λ -termes avec un sous-terme distingué. Sur les λ -termes pointés, on définit la réduction suivante :

$$\begin{array}{ll}
C[\underline{M_1 M_2}] & \rightarrow C[\underline{M_1 M_2}] \\
C[\underline{\lambda x. M}] & \rightarrow C[\underline{\lambda x. M}] \quad (\text{s'il y a un pré-redex } (\lambda x, N)) \\
C[x_i] & \rightarrow C[\underline{M}] \quad (\text{si } (\lambda x, M) \text{ est un pré-redex}) \\
C[\underline{M_1 + M_2}] & \rightarrow C[\underline{M_1 + M_2}] \\
C[\underline{M_1 + M_2}] & \rightarrow C[\underline{M_1 + M_2}]
\end{array}$$

Il est alors facile d'établir une correspondance directe entre les réductions de la KAM et celles sur les λ -termes pointés. Il est aussi facile de voir que cette réduction est une décomposition de la réduction linéaire de tête, qui internalise la phase de recherche de l'occurrence de tête. \square

3.3.3 Machine à pointeurs

La PAM semble avoir été formulée pour la première fois en 1988, indépendamment de toute idée sémantique et en parallèle d'idées comparables de Coquand [Coq95]. La relation avec la sémantique de jeux (notamment la variante HO/N) [DHR96a] a été comprise plus tard. Des variantes ont également été étudiées par Curien et Herbelin [CH98].

Justifions l'introduction de la PAM par la remarque suivante : une séquence d'exécution de la KAM peut être décomposée en alternances de deux phases distinctes. La première est une phase de sélection d'une occurrence de tête, et est composée de transitions (*push, pop, +*). La seconde phase est une phase de substitution, et est composée d'une unique transition (*call*) : c'est cette dernière qui effectue finalement la substitution. L'idée de la machine à pointeurs (PAM) est de ne garder que ces étapes de substitution et de se débarrasser de toutes les coûteuses étapes d'entretien de l'environnement. Chaque état de la PAM correspondra à un état de la KAM avec une occurrence de variable en tête, et chaque transition correspondra à une transition (*call*) de la KAM, suivie de toute la série de transitions ramenant une occurrence de variable en position de tête. A chaque étape, on garde trace de l'occurrence de tête et du terme par lequel cette occurrence doit être substituée. Chaque occurrence est également munie d'un *pointeur* identifiant l'abstraction par laquelle elle est liée. Ainsi, on parvient à simuler la réduction linéaire de tête sans jamais construire de fermeture ou d'environnement.

Pour présenter la machine à pointeurs il va falloir introduire des éléments de terminologie sur les termes, inspirés de [DR]. Notons qu'une partie significative de cette terminologie est empruntée à la sémantique de jeux, mais les intuitions venant de la sémantique de jeux ne s'appliquent vraiment que dans un cas très précis d'exécution de la PAM : il s'agit de ce qu'on appellera une *situation de jeux*. Plus formellement, la PAM sera dans une situation de jeux lorsqu'elle sera exécutée sur un terme de la forme UV , où U et V sont des arbres de Böhm, c'est à dire des termes β -normaux et η -longs. Les définitions de cette section s'appuieront sur des exemples pris en situation de jeux, afin que l'intuition venant des jeux puisse s'appliquer.

Arguments dorsaux d'un terme. Soit M un terme. Son ensemble d'*arguments dorsaux* $dor(M)$ est défini comme suit :

$$\begin{aligned} dor(T_1T_2) &= \{T_2\} \cup dor(T_1) \\ dor(\lambda x.T) &= dor(T) \\ dor(T_1 + T_2) &= dor(T_1) \cup dor(T_2) \\ dor(x_i) &= \emptyset \\ dor(\mathfrak{X}_i) &= \emptyset \end{aligned}$$

L'idée est qu'on parcourt toute la *chaîne dorsale* de M , *i.e.* la chaîne passant toujours à gauche des applications, et qu'on identifie tous les sous-termes en position d'argument à cette chaîne. Par exemple, les arguments dorsaux de $T = (\lambda x.x_0(\lambda y.y_0\mathfrak{X}_0))\mathfrak{X}_1$ sont $\lambda y.y_0\mathfrak{X}_0$ et \mathfrak{X}_1 .

Justification. Si T est un terme et z_0 une occurrence de variable dans T , alors on note T_{z_0} le sous-terme maximal de T qui admet z_0 comme occurrence de tête. Si M est un argument dorsal de T_{z_0} , on dit que z_0 *justifie* M . Dans l'exemple ci-dessus, x_0 justifie ses deux arguments dorsaux et y_0 justifie \mathfrak{X}_0 . La notion de justification est inspirée de la sémantique de jeux. Considérons une situation de jeux :

$$T = (\lambda x.x_0M)N$$

Ici, M et N sont supposés être des arbres de Böhm typés de Λ . Alors, x_0 justifie M et N . Toutes les autres occurrences de variables ne justifient que ses arguments au sens propre, c'est-à-dire que y_0 ne justifie U que si y_0U est un sous-terme de T .

Arguments. Soit T un terme, et x_0 une occurrence de variable dans T . Les *arguments* de x_0 sont les sous-termes de T justifiés par x_0 qui ne font pas partie d'un pré-redex de T_{x_0} . La *chaîne d'arguments* d'une occurrence de variable est la liste de ses arguments, ordonnée de gauche à droite. On la note $arg(x_0)$. Cette liste d'arguments nous permettra de parler de façon non ambiguë du i -ème argument d'une occurrence x_0 . On parlera aussi de l'*arité* de x_0 pour désigner la longueur de sa chaîne d'arguments. Considérons par exemple le terme

$$T = (\lambda z.(\lambda xy.x_0N_1)N_2)N_3N_4N_5$$

Ici, tous les N_i sont justifiés par x_0 . Clairement, N_1 est argument de x_0 puisqu'il est son argument au sens propre. Par contre, N_2 n'est pas argument, il fait en effet partie d'un pré-redex $(\lambda x.N_2)$. De même, N_3 fait partie d'un pré-redex $(\lambda z.N_3)$ et N_4 d'un pré-redex $(\lambda y.N_4)$. Le dernier argument de x_0 est donc N_5 , d'où on déduit $arg(x_0) = [N_1, N_5]$. x_0 est donc d'arité 2.

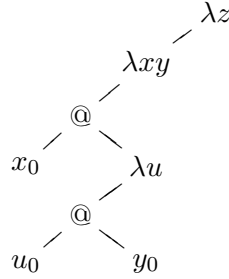
Dans une situation de jeux $T = UV$, les arguments d'une occurrence de variable sont *toujours* ses arguments au sens propre. De plus, si T est typable, chaque occurrence de variable ne peut avoir qu'un unique argument. Cette observation permettra de simplifier drastiquement la machine à pointeurs dans le cas des situations de jeux.

Liason. Soit T un terme. L'ensemble des *abstractions dorsales* de T est défini par induction sur T :

$$\begin{aligned}\lambda(T_1 T_2) &= \lambda(T_1) \\ \lambda(\lambda x.T) &= \{\lambda x\} \cup \lambda(T) \\ \lambda(T_1 + T_2) &= \lambda(T_1) \cup \lambda(T_2) \\ \lambda(x_i) &= \emptyset \\ \lambda(\mathbf{x}_i) &= \emptyset\end{aligned}$$

$\lambda(T)$ est similaire à $\lambda_h(T)$ à ceci près qu'on ne retire pas les abstractions en position de pré-redex. Soit maintenant x_0 une occurrence de variable dans T . Le *sous-terme lieur* de x_0 est défini comme le sous-terme M maximal de T tel que $\lambda x \in \lambda(M)$.

Considérons par exemple le terme $T = (\lambda z.(\lambda xy.x_0(\lambda u.u_0 y_0)))$. Observons son arbre de syntaxe :



Pour trouver le terme lieur de x_0 dans cette représentation, il suffit de partir de x_0 et de remonter le long de l'arbre. On tombe donc sur T tout entier. On fait de même pour y_0 , mais une fois arrivés à λu on a toujours pas croisé de λy . On prend donc à gauche, et on tombe sur T tout entier. Finalement, le lieur de u_0 est le sous-terme $\lambda u.u_0 y_0$.

Dans une situation de jeux, le lieur d'une occurrence x_0 est *toujours* l'unique sous-terme commençant par l'abstraction $\lambda x \dots$ liant x_0 .

Chaîne de sous-termes. Soit T un terme et x_0 une occurrence de variable dans T . La *chaîne de sur-termes* de x_0 est la chaîne (A_0, \dots, A_k) de sur-termes de T tels que $A_0 = T$, pour tout i , A_{i+1} est un argument dorsal de A_i et x_0 est une occurrence de tête de A_k . Cette chaîne est unique, puisqu'elle correspond exactement au chemin dans T depuis la racine vers x_0 , qui est unique puisque T est un arbre. En conséquence, x_0 est liée dans T si et seulement si son sous-terme lieur est présent dans sa chaîne de sur-termes.

Définition 3.3.4. Soit M un terme. Un état de la PAM sur M est une séquence de triplets $(h_1, A_1, p_1), \dots, (h_n, A_n, p_n)$ où chaque h_i est une occurrence de variable de T , chaque A_i est un sous-terme de T , et $0 \leq p_i \leq i$. De plus, cette séquence doit satisfaire les conditions suivantes :

- (1) $h_1 \in \text{hoc}(T)$ et pour tout $i \geq 1$, $h_{i+1} \in \text{hoc}(A_i)$;
- (2) Pour tout $i \geq 1$, A_i est justifié par h_{p_i} .

Pour donner sens à (2) au cas où $p_i = 0$, on pose que les sous-termes justifiés par h_0 sont les mêmes que ceux justifiés par h_1 . De même, on posera par convention $A_0 = T$ pour tout le reste de la section.

Encore une fois, l'intuition est plus simple dans le cas d'une situation de jeux. Si on a $T = UV$, alors on commence typiquement avec h_1 occurrence de tête de U et $A_1 = V$. Pour passer de l'étape i à l'étape $i + 1$, on choisit une occurrence de tête h_{i+1} dans A_i , puis on va identifier à l'aide des pointeurs le sous-terme A_{i+1} qui lui correspond. Dans ce cas, on se rend compte qu'une fois sur deux, l'occurrence de tête est dans U , le reste du temps elle est dans V : on a *alternance de la polarité*.

Dans le cas général, on peut voir l'état de PAM $(h_1, A_1, p_1), \dots, (h_n, A_n, p_n)$ sur T comme une description du terme T' qu'on obtient de T après avoir effectué les n étapes de réduction linéaire correspondant aux substitutions $(h_1, A_1), \dots, (h_n, A_n)$. Cependant, la PAM n'effectue jamais vraiment ces substitutions. A la place, on peut voir les A_i et les h_i comme un registre des *copies* des sous-termes et occurrences de variables de T qui apparaissent dans le réduit T' . Pour que cette information soit suffisante pour calculer les transitions suivantes, on entretient aussi les *pointeurs* p_i pour enregistrer comment ces copies s'organisent entre elles. Comme la définition d'état le montre, il suffit en fait de se souvenir, pour chaque copie de sous-terme A_i , par quelle copie d'occurrence de variable elle est justifiée.

Vue. Dans le cas général, la définition d'un état de la PAM force une propriété structurelle qui est fondamentale pour l'exécution de la PAM, c'est la propriété de *vue*.

Lemme 3.3.2. *Soit $S = (h_1, A_1, p_1) \dots (h_n, A_n, p_n)$ un état de la PAM pour un terme T , et soit h une occurrence de tête de A_n . On pose $\phi_S(i) = \max(p_i - 1, 0)$. Soit k le plus petit entier tel que $\phi_S^k(n) = 0$. Alors $\{A_{\phi^k(n)}, \dots, A_{\phi(n)}, A_n\}$ est la chaîne de sur-termes de h .*

Démonstration. Pour commencer, on remarque que k est bien défini puisque pour tout i , $\phi_S(i) < i$. On a donc $A_{\phi_S^k(n)} = T$ et h est occurrence de tête de A_n . De plus, pour tout i , A_i est un argument dorsal de $A_{\phi_S(i)}$; en effet A_i est justifié par h_{p_i} qui est lui-même une occurrence de tête de A_{p_i-1} donc A_i est argument dorsal de A_{p_i-1} . Or, si $p_i \neq 0$, $p_i - 1 = \phi_S(i)$. Si $p_i = 0$, alors A_i est justifié par $h_0 = h_1$ qui est occurrence de tête de $A_0 = T$, et donc est bien argument dorsal de $A_{\phi_S(i)} = A_0$. \square

La principale conséquence de ce lemme est que, comme on l'a vu un peu plus haut, le sous-terme lieu d'une occurrence h doit apparaître exactement une fois dans la chaîne de sur-termes de h . En fait, cet ensemble $\{A_{\phi^k(n)}, \dots, A_{\phi(n)}, A_n\}$ correspond à la notion de *P-vue* en sémantique de jeux⁶. Dans une situation de jeux, la vue d'une occurrence de variable correspond exactement aux sous-termes abstractions qui peuvent éventuellement lier (au sens propre) cette occurrence. Cette définition est aussi cruciale pour la définition des transitions de la machine à pointeurs. En particulier, on verra que le choix des pointeurs préserve en permanence la *visibilité*.

Initialisation de la PAM. On veut initialiser la PAM sur un terme M . Supposons pour commencer que M est en forme normale pour la réduction linéaire de tête, c'est-à-dire qu'il n'y a pas de redex $(\lambda x, x_0, N)$. Dans ce cas, le seul état de PAM correct sur M est la séquence vide : la machine ne s'initialise pas. Autrement, on suppose donné l'un de ces redex et on définit l'*initialisation de la PAM sur M avec le redex $(\lambda x, x_0, N)$* comme suit :

- On fixe $h_1 = x_0$;
- On fixe $A_1 = N$;
- On fixe $p_1 = 0$.

⁶Plus précisément, cet ensemble correspond aux *coups Opposants* de la *P-vue*.

Récupération d'argument. Par souci de clarté, commençons par décrire en détail une des procédures nécessaires à la transition. Le problème est le suivant : on a un état de PAM $(h_1, A_1, p_1), \dots, (h_n, A_n, p_n)$ sur un terme T . Par convention, notons T_j le terme obtenu de T par j étapes de réduction linéaire de tête, en effectuant les substitutions indiquées par l'état de la PAM. On identifie une copie de sous-terme A_i , et on veut récupérer son k -ième argument. Plus précisément on considère la forêt d'abstractions $\lambda_h(A_i)$ supposée de profondeur au moins k , et on note λx l'une des abstractions de profondeur k ⁷. le k -ième argument de A_i , s'il existe, sera le sous-terme M de T_n tel que $(\lambda x, M)$ sera un pré-redex de T_n . Toute la difficulté est de retrouver M sans construire T_n , juste en utilisant les pointeurs. On procède comme suit :

Étape 1. On note que A_i a remplacé h_i à la i -ème étape (virtuelle) de réduction, donc on commence par regarder les arguments de h_i dans T . Si h_i est d'arité au moins k dans T , on récupère son k -ième argument noté B et on renvoie le couple (B, i) , en notant que h_i justifie B . Les deux éléments de ce couple seront importants pour définir la transition de la PAM. Si h_i est d'arité $a < k$, on passe à l'étape 2.

Étape 2. Si on est ici, c'est que h_i n'a pas suffisamment d'arguments dans T . Mais il peut en avoir suffisamment dans T_n ! En effet, l'arité d'une occurrence de variable peut augmenter lorsqu'elle est copiée. Par exemple, considérons le terme $T = (\lambda x.x_0 AB)(\lambda y.y_0 CD)$. Ici, y_0 est d'arité 2. Mais après une étape de substitution linéaire, on obtient $T' = (\lambda x.(\lambda z.z_0 CD) AB)(\lambda y.y_0 CD)$, et z_0 est une copie de y_0 qui est maintenant d'arité 3. Si $i = 1$, alors on ne trouvera plus d'argument et la machine s'arrête. Sinon, on note qu'on peut trouver des arguments supplémentaires à h_i dans les arguments de h_{i-1} , puisque h_i est occurrence de tête de A_{i-1} qui a remplacé h_{i-1} à la $i - 1$ -ième étape. Seulement, si on cherchait le k -ième argument de h_i on va maintenant chercher le $(k - a + l)$ -ième argument de h_{i-1} , où a est l'arité de h_i et l est la profondeur de la forêt d'abstractions de A_{i-1} . On recommence donc à l'Étape 1 avec ces nouvelles données.

Cette procédure peut sembler compliquée, mais elle est le prix à payer pour ne pas avoir à entretenir d'environnements ou à faire de réduction linéaire de tête. Essentiellement, l'étape 2 arrive quand le terme examiné n'est pas complètement η -expansé. Heureusement, cela n'arrivera pas dans une situation de jeux, ce qui permettra une simplification drastique de la procédure.

Transitions de la PAM. Supposons qu'on ait un état $(h_1, A_1, p_1), \dots, (h_n, A_n, p_n)$ de la PAM sur un terme M . Alors, on construit un état successeur de la façon suivante :

- (1) On choisit une occurrence de tête $h_{n+1} \in \text{hoc}(A_n)$. Si h_{n+1} est un démon, on s'arrête.
- (2) On commence par trouver le lieu de h_{n+1} . Si h_{n+1} est libre, aucune transition ne s'applique. Sinon, on sait par le Lemme 3.3.2 qu'il est lié dans sa vue $\{A_{\phi^k(n)}, \dots, A_{\phi(n)}, A_n\}$, c'est-à-dire qu'il existe j tel que $A_{\phi^j(n)}$ lie h_{n+1} . On pose alors $k = \phi^j(n)$.
- (3) On sait donc que A_k est le sous-terme lieu de h_{n+1} , c'est-à-dire qu'il contient l'abstraction λx dans son épine dorsale. Deux cas sont maintenant envisageables : est-ce que A_k attend véritablement x en argument, ou bien y'a-t-il un pré-redex $(\lambda x, B)$ à l'intérieur de A_k ?
 - (a) Premier cas, il y a bien un pré-redex $(\lambda x, B)$ dans A_k . Dans ce cas, h_{n+1} est sur le point d'être (virtuellement) substitué par B , on pose donc $A_{n+1} = B$. Si $k = 0$, on pose $p_{n+1} = 0$. Sinon, on pose $p_{n+1} = n + 1$ car h_{n+1} justifie bien B , et on satisfait la

⁷Laquelle n'est pas vraiment important, les abstractions de même profondeur participent en effet toutes aux mêmes pré-redex.

propriété (2) de l'état de la PAM.

- (b) Second cas, il n'y a pas de tel pré-redex. λx apparaît donc dans la forêt d'abstractions $\lambda_h(A_k)$. Supposons qu'elle arrive en profondeur k : cela signifie que h_{n+1} sera substitué par le k -ième argument de A_k . Mais A_k a (virtuellement) remplacé l'occurrence h_k a la k -ième étape du calcul, ce qui nous intéresse donc est en fait le k -ième argument de h_k . On lance donc la procédure de récupération d'argument avec ces données. Soit la machine s'arrête, soit on obtient un couple (B, i) , on pose alors $A_{n+1} = B$ et $p_{n+1} = i$, qui satisfont bien la condition (2) de l'état de la PAM.

Remarque. Il y a une subtile différence entre la PAM présentée ici et celle de [DR]. Ici, les p_i peuvent avoir une valeur nulle alors que dans [DR], la valeur minimale est 1. Il se trouve que cette variante est nécessaire pour avoir la relation la plus directe possible avec les jeux dans les situations de jeux. Plus précisément, cette variante donne l'alternance Joueur/Opposant dans les situations de jeux, comme nous expliquerons plus tard. D'un point de vue technique cela ne change pas grand chose, en effet la différence est seulement dans la situation initiale et les vues restent inchangées.

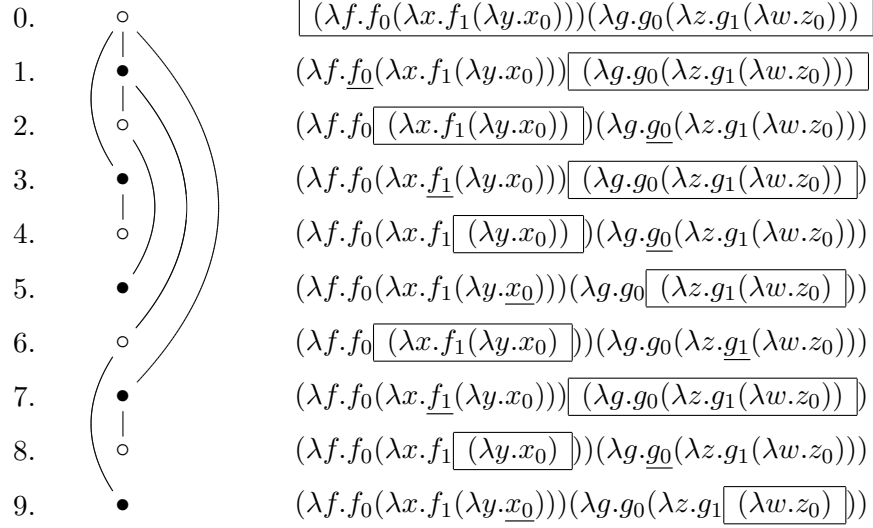
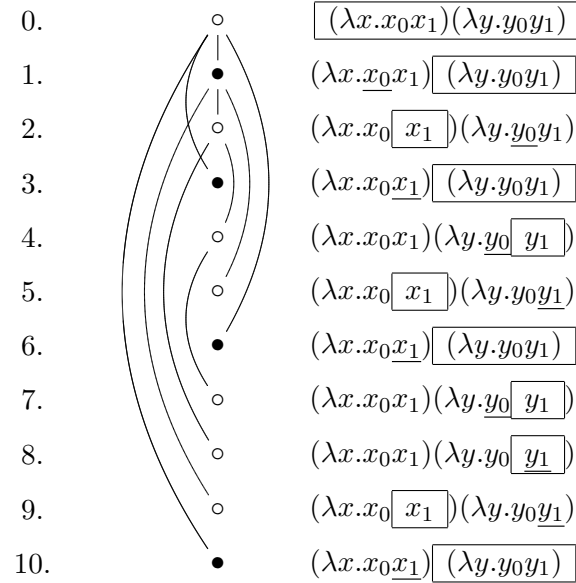
Exemple 1. Notre premier exemple sera une situation de jeux. On rappelle le fameux terme de Kierstead $K_x = \lambda f.f(\lambda x.f(\lambda y.x))$. Considérons l'exécution de la PAM sur K_x appliqué à lui-même, soit après α -renommage le λ -terme suivant :

$$T = (\lambda f.f_0(\lambda x.f_1(\lambda y.x_0)))(\lambda g.g_0(\lambda z.g_1(\lambda w.z_0)))$$

L'exécution de la machine à pointeurs sur ce terme est donnée en Figure 3.1. Par convention, on va à chaque étape i souligner h_i et encadrer A_i . On représentera les p_i graphiquement, en représentant par \bullet les coups «Joueur», où l'occurrence de tête est dans le terme de gauche, et par \circ les coups «Opposant», où l'occurrence de tête appartient au terme de droite. Cette représentation devrait sembler familière aux lecteurs habitués à la sémantique des jeux : sur ce genre de termes où il n'y a pas de coupure interne (juste une coupure principale entre deux termes en forme normale), la PAM calcule précisément l'interaction innocente entre deux stratégies, et la suite de pointeurs admet de bonnes propriétés comme l'alternance Joueur/Opposant et la visibilité. À la fin, la machine s'arrête parce qu'on aurait $h_{10} = z_0$, $p_{10} = 5$ puisque la seule abstraction de z dans la vue de 9 est au cinquième pas d'exécution, mais poser $A_{10} = \text{arg}(h_5)$ est impossible vu que $h_5 = x_0$ n'a pas d'argument dans T .

Exemple 2. Donnons maintenant un exemple hors des situations de jeux. C'est l'exemple de $(\lambda x.xx)$ appliqué à lui-même. On garde la convention de noter \bullet les coups «Joueur» et \circ les coups «Opposant», mais comme on le verra, le fait d'avoir un terme non η -expansé casse toutes les propriétés d'alternance et de visibilité de la suite de pointeurs. On peut voir le résultat sur la Figure 3.2.

Il est assez intéressant de comparer les pointeurs de la Figure 3.2 avec ceux de la Figure 3.1. En effet, le terme K_x est le début de l'arbre de Böhm infini de $\lambda x.xx$, et de ce fait, la Figure 3.1 présente le début de l'interaction qu'on aura en sémantique de jeux entre $\lambda x.xx$ et lui-même. Cette comparaison permet de réaliser que pour passer de la Figure 3.1 aux six premiers coups de la Figure 3.2, il suffit d'oublier certains coups. Ce sont les coups «copycat» correspondant à l' η -expansion, et il est assez naturel que leur oubli amène à perdre des propriétés comme l'alternance des joueurs ou la visibilité. On peut aussi remarquer que la recherche d'arguments de la transition 2.b revient à jouer silencieusement ces coups copycat.

FIG. 3.1 – L'exécution de la PAM sur $K_x K_x$.FIG. 3.2 – L'exécution de la PAM sur $\delta\delta$.

Correction de la PAM. On va relier l'action de la PAM à la réduction linéaire de tête.

Définition 3.3.5. Soit $S = (h_1, A_1, p_1), \dots, (h_p, A_p, p_p)$ un état de la PAM. Considérons toutes les extensions propres (i.e. on effectue au moins une transition) de S par les transitions définies plus haut. Pour chacune de ces extensions non triviales $(h_1, A_1, p_1), \dots, (h_n, A_n, p_n)$, on extrait la chaîne $(h_{p+1}, A_{p+1}), \dots, (h_n, A_n)$. L'arbre résiduel de substitutions pour S est défini comme l'arbre de ces séquences.

L'arbre des substitutions pour la PAM d'un terme M est défini comme l'arbre résiduel de substitutions pour la PAM vide sur M . L'arbre dirigé sur A est le sous-arbre de l'arbre des substitutions pour la PAM de M où l'occurrence de tête pour la première étape est nécessairement choisie dans A .

La relation entre la PAM et la réduction linéaire de tête dirigée va être faite par l'intermédiaire du lemme suivant :

Lemme 3.3.3. Soit M un terme, supposons que $M \rightsquigarrow^{LHR} M'$ par la réduction de $(\lambda x, x_0, N)$. De même, on effectue une étape de PAM sur M pour obtenir l'état $S = (x_0, N, 0)$. Alors, l'arbre résiduel de substitutions de S est égal à l'arbre des substitutions de la PAM de M' dirigé sur N .

Démonstration. On va montrer que toute exécution de la PAM sur M donne lieu à une exécution équivalente de la PAM sur M' , donnant en particulier le même arbre de substitutions. On montrera ensuite la réciproque.

Pour toute exécution de PAM $S = (h_1, A_1, p_1), \dots, (h_n, A_n, p_n)$ sur M , il existe une exécution de la PAM $S' = (h'_1, A'_1, p'_1), \dots, (h'_{n-1}, A'_{n-1}, p'_{n-1})$ sur M' dirigée sur N telle que pour tout i , $h'_i = h_{i+1}$, $A'_i = A_{i+1}$ et si $p_{i+1} > 1$, $p'_i = p_{i+1} - 1$ et si $p_{i+1} \in \{0, 1\}$, $p'_i \in \{0, 1\}$.

On va construire par induction sur n l'exécution de la PAM sur M' correspondant à S . D'abord, on initialise S' avec $(h_2, A_2, 0)$, ce qui est bien une initialisation dirigée sur N comme requis. Supposons maintenant qu'on a construit $S'_i = (h'_1, A'_1, p'_1), \dots, (h'_i, A'_i, p'_i)$. Si $i = n - 1$, la preuve est finie. Sinon,

- On sait que $h'_{i+1} \in \text{hoc}(A'_i)$ puisque la transition correspondante est correcte sur S .
- On calcule la vue $V = \{A_{\phi_S^k(i+1)}, \dots, A_{\phi_S(i+1)}, A_{i+1}\}$ de $h_{i+2} = h'_{i+1}$ dans S et sa vue $V' = \{A'_{\phi_{S'}^k(i+1)}, \dots, A'_{\phi_{S'}(i+1)}, A'_{i+1}\}$ dans S' . Par le Lemme 3.3.2, V est la chaîne de sur-termes de h_{i+2} dans M , et V' sa chaîne de sur-termes dans M' . Puisque M' est obtenu de M par substitution d'une occurrence de tête, on sait que ces deux chaînes doivent coïncider sauf pour le premier élément, qui est respectivement M et M' . Par conséquent, le lieu de h_{i+2} est le même pour les deux machines, à moins que ce soit le terme tout entier auquel cas c'est respectivement M et M' . On raisonne maintenant par cas sur le choix de la transition pour S .

- 3.a Si le lieu de h_{i+2} pour S est $A_0 = M$ et qu'il contient un pré-redex $(\lambda x, B)$ pour h_{i+2} , alors le pré-redex existe aussi nécessairement dans M' , puisqu'en effet la réduction linéaire de tête n'efface jamais de pré-redex. La transition donne alors $A_{i+2} = A'_{i+1} = B$ et $p_{i+2} = p'_{i+1} = 0$, ce qui satisfait bien la propriété attendue de S' . Si le lieu de h_{i+2} dans S est un sous-terme A_j dans S , alors dans S' ce sera $A'_{j-1} = A_j$. Comme on est dans le cas 3.a, c'est que A_j contient un pré-redex $(\lambda x, B)$ liant h_{i+2} . Dans ce cas, la transition de la machine ajoute $(h_{i+2}, B, i + 2)$ à S et $(h'_{i+1}, B, i + 1)$ à S' , ce qui correspond bien à la définition de S' .

3.b La machine S lance la procédure de récupération d'argument. Là encore, plusieurs sous-cas sont possibles. Si le lieu de h_{i+2} est en fait A_1 , que A_1 n'a pas de redex intérieur et que l'abstraction λy liant h_{i+2} est à profondeur k dans $\lambda_h(A_{i+2})$, alors S recherche le k -ième argument de $h_1 = x_0$ dans M , s'il existe. Mais le k -ième argument de x_0 dans M est exactement le B éventuel tel que $(\lambda y, B)$ est un pré-redex dans M' . Pour résumer, la transition sur S choisit A_{n+2} le k -ième argument de x_0 , et $p_{i+2} = 1$. Par contre pour S' le lieu de h'_{i+2} est A'_0 puisque A_1 ne figure plus dans la vue. Puisque B est le k -ième argument de x_0 dans M , c'est précisément qu'il existe un pré-redex $(\lambda y, B)$ dans $M' = A'_0$. La machine S' choisit donc $A'_{i+1} = B = A_{i+2}$ et $p'_{i+1} = 0 = p_{i+2} - 1$, ce qui satisfait bien la propriété attendue. Autre sous-cas, si h_{n+2} admettent cette fois le même lieu A_j aussi bien dans S que dans S' , et que A_j ne contient pas de pré-redex liant h_{n+2} . Dans ce cas on lance la recherche d'arguments des deux côtés. Si la recherche sur S n'atteint jamais h_1 , alors la recherche se passe exactement dans S' , avec le même résultat, vu que les données sont les mêmes. Comme on sait qu'on n'a pas commencé la recherche par h_1 (le lieu de h_{i+2} étant supposé différent de A_1), on sait qu'à l'étape précédente, on recherchait un certain r -ième argument de $h_2 = h'_1$. Mais en fait, il suffit de remarquer que les arguments de h'_1 dans M' sont exactement ceux de h_2 dans M suivis de ceux de x_0 dans M ! Soit a l'arité de h_2 dans M , et soit l la profondeur de la forêt d'abstractions de A_2 . Dans S , la recherche du r -ième argument de h_2 se réduit à la recherche du $r - a + l$ -ième argument de x_0 : on récupère exactement le même sous-terme que le r -ième argument de h_2 dans M' . Soit B ce sous-terme, à cette étape, on a donc ajouté le triplet $(h_{i+2}, B, 1)$ à S et $(h_{i+2}, B, 1)$ à S' , ce qui valide la propriété attendue.

De même, pour toute exécution de la PAM $S' = (h'_1, A'_1, p'_1), \dots, (h'_{n-1}, A'_{n-1}, p'_{n-1})$, on peut construire une exécution S équivalente, dans le même sens qu'écrit plus haut. Les détails de la preuve sont exactement les mêmes que pour le cas précédent, on ne les répète pas.

Finalement, on remarque que pour tout état S , l'état S' construit à la même chaîne de substitutions et réciproquement, ce qui prouve bien l'énoncé du lemme. \square

Théorème 3.3.4. *La PAM est correcte : si M est un terme, l'arbre des substitutions dirigées de M pour la réduction linéaire de tête est identique à son arbre de substitutions pour la PAM.*

Démonstration. Définissons une relation \mathcal{R} entre les couples (M, N) , où N est un sous-terme de M , et les états de la PAM en disant que $(M, N)\mathcal{R}S$ si et seulement si l'arbre des substitutions de la PAM sur M dirigée sur N est égal à l'arbre résiduel de substitutions de S . On raisonne ensuite par coinduction, en utilisant \mathcal{R} comme invariant.

Clairement, on a $(M, M)\mathcal{R}S_M$ où S_M est l'état de PAM vide sur M . Supposons maintenant qu'on aie des termes U et V et un état de PAM S , avec $(U, V)\mathcal{R}S$. Supposons qu'on aie $U \rightsquigarrow^{LHR} U'$ par la substitution (x_0, W) , où $x_0 \in \text{hoc}(V)$. On utilise alors le fait que $(U, V)\mathcal{R}S$: cela implique en effet qu'on peut faire la transition correspondante (x_0, W) dans la PAM pour obtenir un état S' . Mais par le Lemme 3.3.3, on a alors encore $(U', W)\mathcal{R}S'$.

De même, si $(U, V)\mathcal{R}S$ et S fait une transition (x_0, W) pour arriver dans S' , on en déduit que U contient un redex de réduction linéaire de tête $(\lambda x, x_0, W)$ avec $x_0 \in \text{hoc}(V)$. On a donc aussi $U \rightsquigarrow^{LHR} U'$ par cette substitution, et le Lemme 3.3.3 nous assure que $(U', W)\mathcal{R}S'$.

L'énoncé du théorème est alors une conséquence du raisonnement ci-dessus par coinduction. \square

La PAM en situation de Jeux. Dans ce paragraphe, on donne finalement quelques précisions sur le fonctionnement de la PAM en situation de jeux. Pour simplifier, on va considérer les interactions entre arbres de Böhm unaires, comme définis par les règles de typage suivantes :

— Arbres de Böhm Unaires —

$$\frac{}{\Gamma \vdash \lambda x. \mathbf{X} : k} \quad \frac{\Gamma, x : k \vdash M_1 : p \quad \dots \quad \Gamma, x : k \vdash M_{|I|} : p}{\Gamma \vdash \lambda x. \oplus_{i \in I} x_i M_i : k + 1}$$

Avec la contrainte pour la seconde règle que pour tout $i \in I$, $(x_i : p + 1) \in \Gamma \cup \{x : k\}$.

Définition 3.3.6. Une situation de jeux pour la PAM est un terme typé de $\Lambda^0 UV$, avec U et V des arbres de Böhm unaires.

Proposition 3.3.5. Soit $T = UV$ une situation de jeux, alors chaque occurrence de variable x_0 de T a un unique argument dans T , noté $\text{arg}(x_0)$.

Démonstration. x_0 est soit dans U soit dans V , dans les deux cas on conclut par induction sur la dérivation de type de l'arbre de Böhm. \square

Supposons maintenant qu'on ait un état de la PAM $(h_1, A_1, p_1), \dots, (h_n, A_n, p_n)$ sur un terme UV en situation de jeux. Essentiellement, cela permet de faire les simplifications suivantes sur la transition de la PAM.

1. On choisit $h_{n+1} \in \text{hoc}(A_n)$. Si h_{n+1} est un démon, on s'arrête.
2. Sinon, on trouve le lieu A_j de h_{n+1} . On pose $p_{n+1} = j$
3. On pose $A_{n+1} = \text{arg}(h_{p_{n+1}})$.

Ces étapes simplifiées sont justifiées par le fait qu'en situation de jeux, on va toujours utiliser la transition 3.b de la PAM, (sauf si le lieu est A_0 , auquel cas on aura tout de même $p_{n+1} = 0$ et $A_{n+1} = V$), et la recherche d'arguments va toujours terminer à sa première étape puisque le terme est complètement η -expansé. On constate que dans ce cadre simplifié, la machine à pointeurs est beaucoup plus efficace. On note aussi qu'en situation de jeux, on aura alternance dans la chaîne de substitutions entre les occurrences de U et celle de V , et chaque joueur pointera dans sa vue.

Cette simplification de la PAM n'est pas restreinte au cas unaire, mais fonctionne aussi dès lors qu'on l'exécute sur UV , avec U et V des termes simplement typés, β -normaux et η -longs. Dans ce cas, on n'a pas unicité de l'argument mais la recherche du k -ième argument d'une occurrence demeure en temps constant : il suffit de regarder les arguments immédiats de l'occurrence.

Chapitre 4

Pointeurs et Normalisation

Sommaire

4.1 Structures de pointeurs	94
4.1.1 Arènes pures et écrasement	94
4.1.2 Définitions et propriétés	94
4.2 Théorèmes de finitude	103
4.2.1 PAM inversée et réduction linéaire de tête	103
4.2.2 Agents, interactions et réalisabilité	106
4.2.3 Approche directe	115
4.3 Conséquences	120
4.3.1 Résultats intermédiaires	120
4.3.2 Stratégies finies, bornées, noethériennes	123

Dans ce chapitre, nous allons introduire et explorer la notion de *structure de pointeurs*. Il y a plusieurs façons d'introduire les structures de pointeurs. Dans le cadre de la sémantique de jeux HO on peut les voir comme le squelette des interactions, c'est-à-dire ce qu'il reste des parties après avoir oublié les étiquettes des coups, et ce aussi bien en appel par nom qu'en appel par valeur. Cette relation est bien visible dans [HHM07], où une partie est définie¹ comme un couple (φ, s) , où φ est une *fonction de pointeurs pairs*², notion très proche des structures de pointeurs, et s est une suite de coups. On peut aussi voir toute structure de pointeurs comme la séquence p_i de l'exécution de la PAM en situation de jeux, comme on verra dans la Section 4.2.1. Cette notion est également apparue plus ou moins explicitement dans de nombreux travaux en logique et sémantique [Cur98, Coq95]. Étant donné l'apparition naturelle de cette notion dans de nombreux contextes différents, il paraît naturel de s'y intéresser.

Dans cette thèse, nous nous intéressons à ces structures de pointeurs parce qu'elles permettent d'apporter une réponse au problème de *dialogue infini* causant les éventuelles ruptures de la totalité par composition. Dans la Section 4.1 on va introduire ces structures dans un cadre de sémantique de jeux, et ensuite en donner une axiomatisation élémentaire. Dans la Section 4.2 on énoncera les théorèmes de finitude permettent de bannir le phénomène de dialogue infini. On donnera ensuite plusieurs preuves de ces théorèmes : d'abord en les reliant à l'exécution de la machine à pointeurs définie précédemment, puis par une preuve sémantique par réalisabilité indépendante de toute syntaxe, et finalement par une approche plus

¹En fait c'est un peu plus compliqué, mais cette décomposition en est une première approximation.

²*parity pointer function* en anglais.

directe due à Coquand [Coc95]. Finalement, en Section 4.3, on tirera un certain nombre de conséquences de ces théorèmes de finitude sur la composition de plusieurs classes de stratégies totales. Les résultats de ce chapitre ont été publiés dans [CH09].

4.1 Structures de pointeurs

4.1.1 Arènes pures et écrasement

Comme introduit plus haut, l'objectif de ce chapitre est de nous permettre de construire des classes de stratégies totales stables par composition les plus larges possible, afin de pouvoir y interpréter divers systèmes logiques. On va présenter ici une première simplification, qui est le reflet sémantique de notre choix d'un λ -calcul unaire dans le chapitre précédent : l'idée va être de considérer des parties se déroulant non plus dans n'importe quelle arène, mais juste dans une arène «unaire», de profondeur infinie. On justifiera que cette simplification permet toujours de traiter le problème dans toute sa généralité.

Dans la catégorie **Vis** des arènes et stratégies visibles, on définit les arènes *pures*³ comme suit. Pour $n \in \mathbb{N}$, soit I_n l'arène suivante :

$$0 \longrightarrow 1 \longrightarrow \cdots \longrightarrow n$$

de longueur n , et I_ω l'arène infinie correspondante. Ce sont les parties sur cette arène I_ω qui correspondront à ce que nous appellerons les *structures de pointeurs*. On montre rapidement comment n'importe quelle partie sur une arène A quelconque peut être plongée dans I_ω . On définit tout d'abord un morphisme de graphe ρ_A de A vers I_n de la façon suivante : chaque nœud x de A de profondeur n est envoyé vers le nœud n lui-même, et chaque arête entre x et y , de profondeurs respectives n et $n+1$, est envoyé vers l'unique arête de I_ω entre n et $n+1$.

Ce morphisme de graphe ρ_A s'étend alors librement en une fonction ρ_A^* sur les parties légales sur A , comme suit :

$$\rho_A^* = \begin{cases} \mathcal{L}_A & \rightarrow \mathcal{L}_{I_\omega} \\ \epsilon & \mapsto \epsilon \\ sa & \mapsto \rho_A^*(s)\rho(a) \end{cases}$$

Où $\rho_A(a)$ pointe au même endroit que le faisait a . Cette possibilité est garantie par le fait que ρ_A est un morphisme de graphe.

Prendre l'image par ρ_A^* revient à *écraser l'arène* : tous les coups sont identifiés à leur profondeur. On se rend compte immédiatement que toutes les propriétés des parties légales ne reposant pas sur l'égalité entre les coups sont préservées. On a par exemple préservation de la longueur de la partie, de la profondeur de ses coups, et de la visibilité. Par contre, les notions qui dépendent de l'égalité entre les coups ne sont en général *pas* préservées. Par exemple, cette opération peut transformer une partie où Joueur joue non-innocemment en une partie où il est parfaitement innocent (voir Figure 4.1), et inversement (voir Figure 4.2). L'intérêt d'une telle simplification est qu'en fait, aucune des hypothèses de notre théorème de finitude ne reposera sur l'égalité entre les coups. L'analyse de I_ω sera donc parfaitement générale.

4.1.2 Définitions et propriétés

Comme expliqué plus haut, on va se contenter d'examiner le problème de finitude pour les parties légales sur l'arène pure I_ω . Il se trouve que ces objets sont en fait relativement

³La terminologie est à relier avec la notion de *type pur* en λ -calcul.

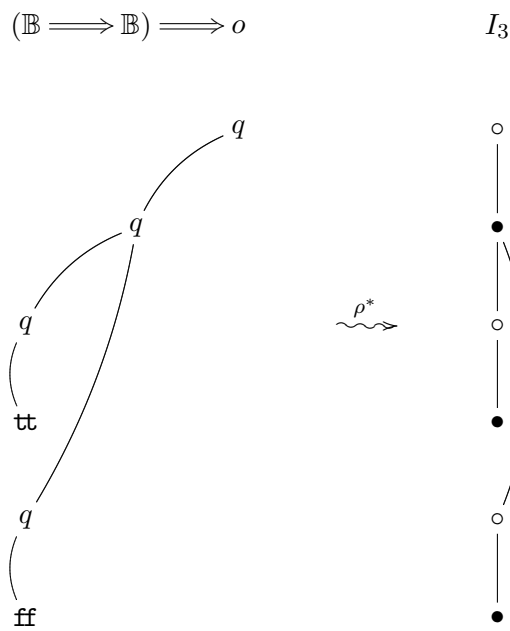


FIG. 4.1 – Comment l'écrasement génère de l'innocence.

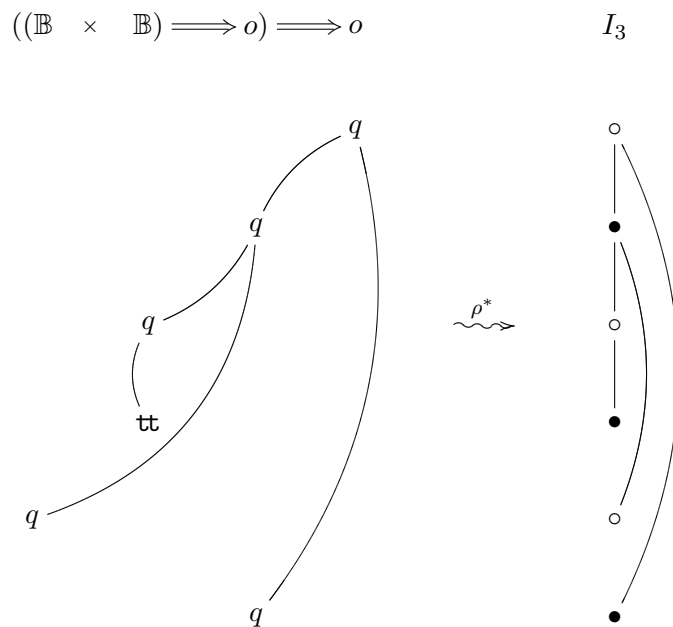


FIG. 4.2 – Comment l'écrasement brise l'innocence.

simples, en ce sens qu'il n'est pas nécessaire d'utiliser tous les outils et le vocabulaire de la sémantique de jeux pour en parler. De plus, comme expliqué au début du chapitre, ils apparaissent naturellement dans des cadres distincts de la sémantique de jeux, par exemple dans les traces d'exécution de la machine à pointeurs. Par conséquent, on va en donner ici une définition élémentaire indépendante de la sémantique de jeux, et on raisonnera pour le moment uniquement sur cette définition élémentaire.

Structures de pointeurs. Dans ce qui va suivre, on utilisera la notation \equiv_2 pour la congruence modulo 2 sur les entiers naturels, et \mathbb{N}_* décrira l'ensemble des entiers naturels privé de 0. L'axiomatisation qu'on va prendre pour les structures de pointeurs est très similaire à celle des fonctions de pointeurs pairs de [HHM07] :

Définition 4.1.1 (Structure de pointeurs). *Une structure de pointeurs est une fonction*

$$\varphi : \mathbb{N} \longrightarrow \mathbb{N} \cup \{\perp\}.$$

À chaque entier on associe l'indice du coup vers lequel il pointe ; \perp est un «pointeur nul», spécifiant le fait que la partie est terminée. De plus, cette fonction doit vérifier les axiomes suivants :

- Partie terminée : pour tout $i \in \mathbb{N}$, si $\varphi(i) = \perp$ alors, pour tout $j > i$, $\varphi(j) = \perp$
- Pointeurs en arrière : pour tout $i \in \mathbb{N}_*$, si $\varphi(i) \neq \perp$ alors $\varphi(i) < i$
- Inversion de la parité : pour tout $i \in \mathbb{N}_*$, $i \not\equiv_2 \varphi(i)$
- Zéro : $\varphi(0) = 0$

On notera Ptr l'ensemble de toutes les structures de pointeurs.

On notera que la valeur en 0 de n'importe quelle structure de pointeurs est supposée être 0. Ce choix n'est pas canonique, n'importe quelle valeur aurait en fait convenu. Contrairement aux conventions habituelles, les coups de l'Opposant sont ici représentés par les valeurs paires et les coups de Joueur par des valeurs impaires.

La définition de structure de pointeurs contient de façon naturelle aussi bien les parties finies qu'infinies. Pour les différencier, on utilise la définition suivante :

Définition 4.1.2. *Soit φ une structure de pointeurs. Le domaine de φ , noté \mathcal{D}_φ , est défini par :*

$$\mathcal{D}_\varphi = \{n \in \mathbb{N}_* \mid \varphi(n) \neq \perp\}$$

On dit que φ est finie si et seulement si \mathcal{D}_φ est fini. Sinon, φ est infinie. Si φ est finie, on définit sa taille $|\varphi|$ comme $|\mathcal{D}_\varphi|$.

Vue. La définition de structure de pointeurs permet de modéliser les notions habituelles de (O/P) -vue. Par commodité, on prend ici une définition un rien différente de ce qu'on a habituellement, même si les notions sous-jacentes sont les mêmes. L'idée est qu'au lieu de définir la vue pour un joueur donné, on va définir la vue d'un coup : elle correspondra à la P -vue si le coup est Joueur (*i.e.* impair) et à la O -vue si le coup est Opposant (*i.e.* pair). Dualelement, on aura la notion de *co-vue d'un coup*.

Définition 4.1.3. *Soit φ une structure de pointeurs. Sa fonction de vue $\lceil \varphi \rceil : \mathcal{D}_\varphi \rightarrow \mathcal{P}(\mathbb{N})$ et sa fonction de co-vue $\lfloor \varphi \rfloor : \mathcal{D}_\varphi \rightarrow \mathcal{P}(\mathbb{N})$ sont calculées de la façon mutuellement récursive*

suivante :

$$\begin{aligned}\lceil\varphi\rceil(0) &= \{0\} \\ \lfloor\varphi\rceil(0) &= \{0\} \\ \lceil\varphi\rceil(i+1) &= \{i+1\} \cup \lfloor\varphi\rceil(i) \\ \lfloor\varphi\rceil(i) &= \{i\} \cup \lceil\varphi\rceil(\varphi(i))\end{aligned}$$

Bien sûr, les notions habituelles de P -vue et d' O -vues peuvent être facilement retrouvées comme suit :

$$\begin{aligned}\lceil\varphi\rceil(i) &= \begin{cases} \lceil\varphi\rceil(i) & \text{si } i \text{ est impair} \\ \lfloor\varphi\rceil(i) & \text{si } i \text{ est pair} \end{cases} \\ \lfloor\varphi\rceil(i) &= \begin{cases} \lceil\varphi\rceil(i) & \text{si } i \text{ est pair} \\ \lfloor\varphi\rceil(i) & \text{si } i \text{ est impair} \end{cases}\end{aligned}$$

Dans ce chapitre, nous lui préférons souvent la première version puisqu'elle permet de raisonner avec les vues sans distinguer de cas selon la polarité du coup courant. Cette simplification est bien illustrée par la définition suivante :

Définition 4.1.4. *Soit φ une structure de pointeurs. On dit que φ est visible si et seulement si :*

$$\forall i \in \mathcal{D}_\varphi, \varphi(i) \in \lceil\varphi\rceil(i)$$

De même, une structure de pointeurs φ est ultimement visible si et seulement si :

$$\exists N \in \mathbb{N}, \forall i \geq N, i \in \mathcal{D}_\varphi \implies \varphi(i) \in \lceil\varphi\rceil(i)$$

Si φ est ultimement visible, on désignera l'entier N à partir duquel elle est visible par son indice de visibilité.

Profondeur. Une notion fondamentale des structures de pointeurs est leur *profondeur*. En particulier, les structures de pointeurs à profondeur finie bénéficieront de nombreuses propriétés intéressantes, notamment pour les questions de finitude.

Définition 4.1.5. *Soit φ une structure de pointeurs. La profondeur de φ est la taille de la plus longue chaîne de pointeurs qu'on peut en extraire. Formellement :*

$$\begin{aligned}p_\varphi(0) &= 0 \\ p_\varphi(i) &= 1 + p_\varphi(\varphi(i)) \\ \text{prof}(\varphi) &= \sup_{i \in \mathcal{D}_\varphi} p_\varphi(i)\end{aligned}$$

Si $\text{prof}(\varphi) = +\infty$, on dira que φ est à profondeur infinie.

Fourches conscientes et mémoire. Restreindre la profondeur est un moyen de contrôler la longueur des structures de pointeurs. Une autre est de contrôler les duplications, ou *fourches*.

Définition 4.1.6. *Soit φ une structure de pointeurs. Une fourche dans φ est un couple $\psi = (\psi_0, (\varphi_i)_{i \in I})$ où I est un segment initial de \mathbb{N}_* , où :*

$$- \forall i \in I, i \in \mathcal{D}_\varphi,$$

$$- \forall i \in I, \varphi(\psi_i) = \psi_0.$$

La taille de la fourche est défini comme le cardinal de l'ensemble I .

Définition 4.1.7. Soit $\varphi \in \text{Ptr}$, soit $\psi = (\psi_0, (\psi_i)_{i \in I})$ une fourche sur ψ . On dit que la fourche ψ est consciente si et seulement si :

$$\forall i \leq j \in I, i \in \ulcorner \varphi \urcorner(j)$$

La mémoire de φ est la borne supérieure (dans $\mathbb{N} \cup \{\infty\}$) de l'ensemble des k tel que φ admet une fourche consciente de taille k . Ainsi, φ est à mémoire finie si et seulement si la mémoire de φ est un entier naturel.

Exemples. Illustrons ces définitions par deux exemples. Premièrement, considérons la structure de pointeurs obtenue par l'interaction entre l'interprétation de $\delta = \lambda x.xx$ et lui-même, représentée en Figure 4.3. Sa profondeur est infinie, il suffit en effet de suivre la branche infinie extérieure pour s'en rendre compte. Par contre, un examen des vues de δ permet de réaliser que sa mémoire est 2, donc finie. Par opposition, la structure de pointeurs dont le début est représenté en Figure 4.4 est à profondeur 2 donc finie, mais à mémoire infinie. Elle a même une *fourche consciente infinie*. Une des façons de voir les théorèmes de finitude de cette section est que ces deux exemples représentent tous les cas de divergence possibles : toute structure de pointeurs visible infinie aura soit une branche de pointeurs infinie, soit une fourche consciente infinie.

Ordre visuel. Si φ est une structure de pointeurs, il est naturel de considérer la relation de visibilité sur \mathcal{D}_φ :

Définition 4.1.8. Soit φ une structure de pointeurs. On définit l'ordre visuel sur \mathcal{D}_φ par :

$$i \preceq_\varphi j \Leftrightarrow (j \in \ulcorner \varphi \urcorner(i) \wedge i \equiv_2 j)$$

La relation stricte correspondante est définie par :

$$i \prec_\varphi j \Leftrightarrow (i \preceq_\varphi j \wedge i \neq j)$$

La plupart du temps, la structure φ sera implicite et ces relations seront juste notées \preceq et \prec .

Cet ordre visuel (on va montrer juste en dessous qu'il définit bien une relation d'ordre) est compatible avec l'inclusion des vues :

Lemme 4.1.1. Soit φ une structure de pointeurs, et $i, j \in \mathcal{D}_\varphi$. Alors, on a les deux propriétés suivantes :

- (1) $j \in \ulcorner \varphi \urcorner(i) \wedge j \equiv_2 i \implies \ulcorner \varphi \urcorner(j) \subseteq \ulcorner \varphi \urcorner(i)$
- (2) $j \in \ulcorner \varphi \urcorner(i) \wedge j \not\equiv_2 i \implies \ulcorner \varphi \urcorner(j) \subseteq \ulcorner \varphi \urcorner(i)$

Démonstration. On montre le résultat par induction sur $i - j$. Premièrement, on note que $i - j$ est toujours un nombre positif, puisque $i \in \ulcorner \varphi \urcorner(j)$. Premier cas, si $i - j = 0$. Plus précisément, on a $i = j$ donc $i \equiv_2 j$ et clairement $\ulcorner \varphi \urcorner(i) = \ulcorner \varphi \urcorner(j)$. Second cas, $i - j = 1$. Dans ce cas on a $i \not\equiv_2 j$, $\ulcorner \varphi \urcorner(i) = \{i\} \cup \ulcorner \varphi \urcorner(j)$ et on a bien (2). Sinon,

$$\ulcorner \varphi \urcorner(i) = \{i, i - 1\} \cup \ulcorner \varphi \urcorner(\varphi(i - 1))$$

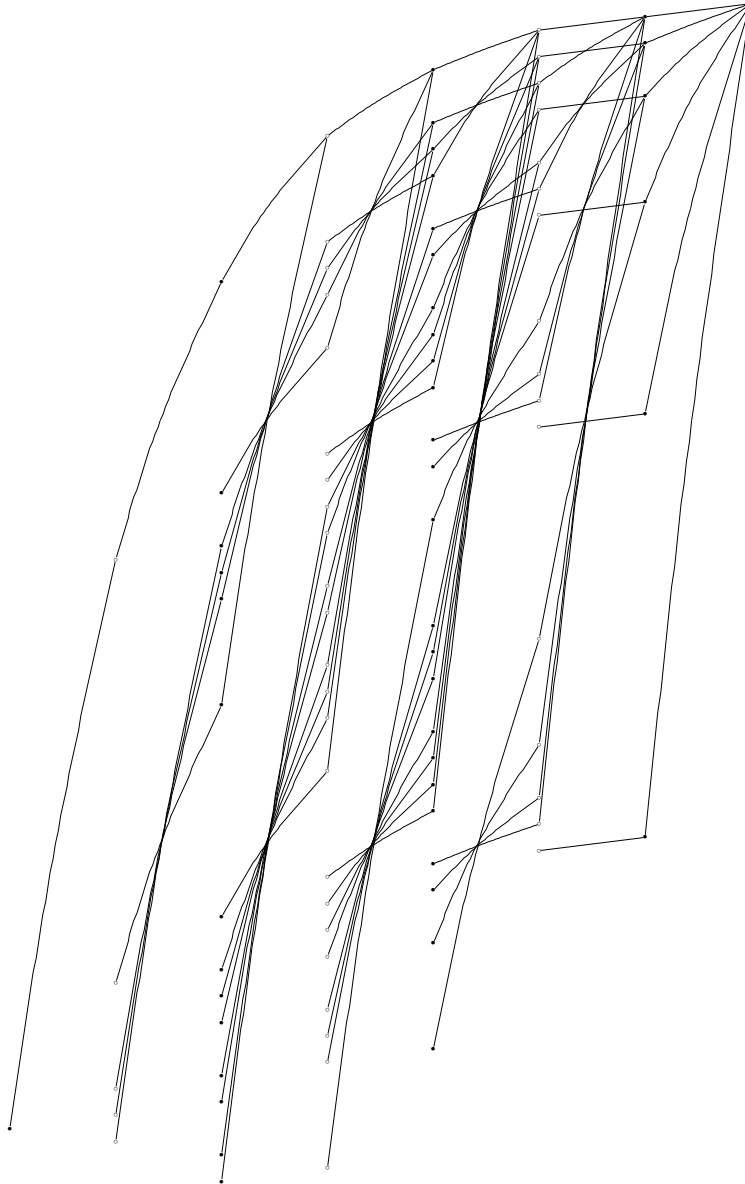


FIG. 4.3 – Le début d’une structure de pointeurs infinie avec une mémoire finie mais une profondeur infinie : la trace de $\delta\delta$.

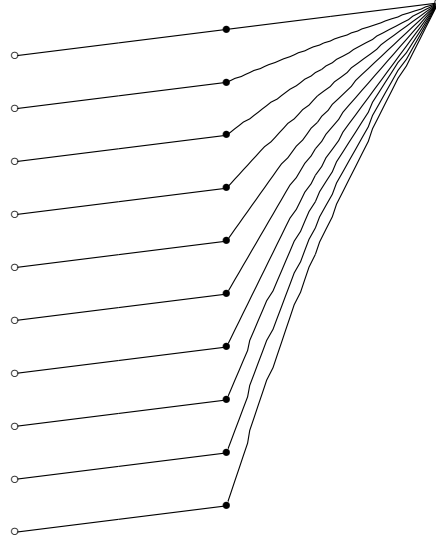


FIG. 4.4 – Le début d’une interaction à profondeur finie mais à mémoire infinie : l’opérateur Y contre l’identité.

Ici, on a toujours nécessairement $j \in \ulcorner \varphi_{\lrcorner}(\varphi(i-1))$, et $j \equiv_2 \varphi(i-1)$ si et seulement si $i \equiv_2 j$. Mais $\varphi(i-1) - j < i - j$ puisque $\varphi(i-1) < i$. Par hypothèse d’induction, on a alors $\ulcorner \varphi_{\lrcorner}(j) \subseteq \ulcorner \varphi_{\lrcorner}(\varphi(i-1))$ si $j \equiv_2 \varphi(i-1)$ et $\ulcorner \varphi^{\lrcorner}(j) \subseteq \ulcorner \varphi_{\lrcorner}(\varphi(i-1))$ sinon. On conclut par transitivité de l’inclusion, en observant que $\ulcorner \varphi_{\lrcorner}(\varphi(i-1)) \subseteq \ulcorner \varphi_{\lrcorner}(i)$. \square

Lemme 4.1.2. *Soit φ une structure de pointeurs, alors \preceq est une relation d’ordre sur \mathcal{D}_{φ} .*

Démonstration. Trois conditions à vérifier :

- *Reflexivité.* Soit $i \in \mathcal{D}_{\varphi}$. Si $i = 0$, $\ulcorner \varphi_{\lrcorner}(i) = \{0\}$ et $0 \in \ulcorner \varphi_{\lrcorner}(0)$. Sinon $\ulcorner \varphi_{\lrcorner}(i) = \{i\} \cup \ulcorner \varphi^{\lrcorner}(i-1)$ et $i \in \ulcorner \varphi_{\lrcorner}(i)$.
- *Antisymétrie.* Notons que si $j \in \ulcorner \varphi_{\lrcorner}(i)$, alors $j \leq i$ (pour l’ordre usuel sur les entiers). Donc si $i \preceq j$ et $j \preceq i$, on a nécessairement $j \leq i$ et $i \leq j$, donc $i = j$.
- *Transitivité.* Application immédiate du Lemme 4.1.1. \square

Lemme 4.1.3. *Soit φ une structure de pointeurs visible à partir de $N \in \mathbb{N}$, alors pour tout $i \in \mathcal{D}_{\varphi}$ tel que $i \geq N$, si $\varphi(i) \neq 0$ on a $i \prec \varphi^2(i)$.*

Démonstration. Puisque φ est visible après N , on a $\varphi(i) \in \ulcorner \varphi_{\lrcorner}(i)$ et comme $\varphi(i) \not\equiv_2 i$ on en déduit par le Lemme 4.1.1 que $\ulcorner \varphi^{\lrcorner}(\varphi(i)) \subseteq \ulcorner \varphi_{\lrcorner}(i)$. Or, on a $\ulcorner \varphi^{\lrcorner}(\varphi(i)) = \{\varphi(i), \varphi^2(i)\} \cup \ulcorner \varphi^{\lrcorner}(\varphi^2(i) - 1)$ donc $\varphi^2(i) \in \ulcorner \varphi^{\lrcorner}(\varphi(i)) \subseteq \ulcorner \varphi_{\lrcorner}(i)$. De plus, on a bien $\varphi^2(i) \equiv_2 i$ et $\varphi^2(i) \neq i$ puisque $\varphi(i) \neq 0$ et par les axiomes des structures de pointeurs, donc $i \prec \varphi^2(i)$. \square

L’intérêt de l’ordre visuel est que ses propriétés (ou plutôt, les propriétés de l’ordre strict \prec) expriment de façon concise les propriétés de profondeur et de mémoire de la structure de pointeurs φ , comme le précise les développements suivants :

Définition 4.1.9. *Une structure de pointeurs φ est bornée si et seulement si l’une des propriétés équivalentes suivantes est satisfaite :*

- (1) $\exists M \in \mathbb{N}, \forall i \in \mathcal{D}_\varphi, |\ulcorner \varphi \urcorner(i)| \leq M,$
 (2) $\exists M \in \mathbb{N}, n_1 \succ n_2 \succ \dots \succ n_p \Rightarrow p \leq M.$

Démonstration. L'équivalence (1) \Leftrightarrow (2) est immédiate. \square

Proposition 4.1.4. *Fixons φ une structure de pointeurs ultimement visible, alors les deux propriétés suivantes sont équivalentes.*

- (1) φ est bornée.
 (2) φ est à profondeur finie et à mémoire finie.

Démonstration. (1) \Rightarrow (2). Soit M la borne sur la taille des vues de φ . Alors :

- Définissons la *longueur* d'une chaîne de pointeurs $x_0 \leftarrow x_1 \leftarrow x_2 \leftarrow \dots \leftarrow x_n$ dans φ comme étant $n + 1$. Considérons les chaînes telles que φ est visible au delà de x_0 . La longueur de telles chaînes de pointeurs dans φ est bornée par M . En effet, on a $x_i \in \ulcorner \varphi \urcorner(x_n)$ pour tout $0 \leq i \leq n$. Si $i \equiv_2 n$, on le sait par itération du Lemme 4.1.3. Sinon, on itère le Lemme 4.1.3 jusqu'à x_{i+1} , puis on conclut par visibilité et par le Lemme 4.1.1. Finalement, toute chaîne de pointeurs dans φ voit sa taille bornée par $M + N$, où N est l'indice de visibilité de φ : il suffit de considérer la chaîne à partir de x_N à partir duquel φ sera nécessairement visible, puisque $x_{i+1} > x_i$ pour tout i . En conclusion, la profondeur de φ est bornée par $M + N$.

- La mémoire de φ est bornée par M . En effet, par définition n'importe quel élément de la fourche consciente voit tous les précédents, sa taille ne peut donc pas dépasser M .

(2) \Rightarrow (1). Soit d la profondeur de φ et m sa mémoire. Considérons $i \in \mathcal{D}_\varphi$, et considérons $\ulcorner \varphi \urcorner(i)$. Cet ensemble peut-être muni d'une structure d'arbre par l'intermédiaire des pointeurs encodés dans φ . On remarque que dans cet arbre, les seuls sommets de degré supérieur à 1 sont de parité opposée à i . C'est clair parce que si on a $j \not\equiv_2 i$ avec $j \in \ulcorner \varphi \urcorner(i)$, alors $\{\varphi(j) + 1, \dots, j - 1\} \cap \ulcorner \varphi \urcorner(i) = \emptyset$, donc si on a j_1 et j_2 avec $j_1 \equiv_2 j_2 \not\equiv_2 i$ et $\varphi(j_1) = \varphi(j_2)$, alors si l'un des deux est dans $\ulcorner \varphi \urcorner(i)$, l'autre ne peut pas y être. Par conséquent, chaque sommet de degré $p \geq 2$ dans $\ulcorner \varphi \urcorner(i)$ est un sommet de polarité opposée à i , donc donne lieu, par définition de $\ulcorner \varphi \urcorner(i)$, à une fourche consciente de taille p dans φ . On en déduit que $\ulcorner \varphi \urcorner(i)$ est un arbre de degré maximal m et de profondeur maximale d , d'où :

$$|\ulcorner \varphi \urcorner(i)| \leq \frac{m^{d+1} - 1}{m - 1}$$

\square

Les deux conditions équivalentes ci-dessus seront les hypothèses du théorème de finitude sur les structures de pointeurs. L'intérêt de l'équivalence présentée en Proposition 4.1.4 apparaît quand on étudie les parties légales sur une arène finie, donc à profondeur bornée. Dans ce cas, la condition de borne sur les vues se transforme en une borne sur la taille des fourches conscientes. Cette dernière condition semble plus intuitive, puisqu'elle correspond très précisément à mettre une borne sur le nombre de duplications autorisées.

Comme on le verra, on pourra également donner une version plus forte du théorème de finitude ou les hypothèses de finitude de la Proposition 4.1.4 seront généralisées en des hypothèses de bonne fondaison. Voici les définitions correspondantes. Ces définitions permettront ensuite de donner une variante de la Proposition 4.1.4 adaptée à ce cas bien fondé.

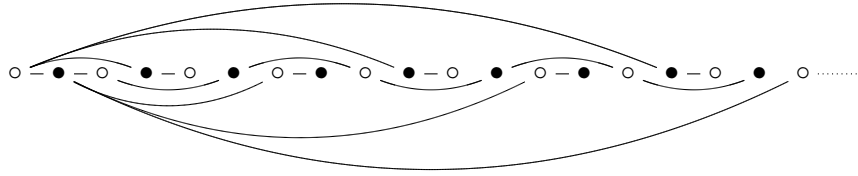


FIG. 4.5 – Deux fourches conscientes infinies opposées.

Définition 4.1.10. Soit φ une structure de pointeurs. On dit que φ est à profondeur bien fondée si et seulement si φ n'admet pas de chaîne de pointeurs infinie, c'est-à-dire s'il n'y a pas de suite $(x_n)_{n \in \mathbb{N}} \in \mathcal{D}_\varphi^{\mathbb{N}}$ telle que pour tout $i \in \mathbb{N}_*$, on ait $\varphi(x_{i+1}) = x_i$. De même, φ est à mémoire bien fondée si et seulement si φ n'admet pas de fourche consciente infinie.

Proposition 4.1.5. Soit φ une structure de pointeurs ultimement visible, les deux propositions suivantes sont équivalentes :

- (1) La relation \prec_φ est bien fondée,
- (2) φ est à profondeur et mémoire bien fondées.

Démonstration. Même raisonnement que pour la Proposition 4.1.4, mais par contraposée. Toute chaîne de pointeurs infinie ou fourche consciente infinie fournit un contre-exemple à (1). De même, d'une suite infinie $(x_n)_{n \in \mathbb{N}}$ on peut construire un arbre infini comme ci-dessus, lequel induit par le lemme de König soit un nœud de degré infini (*i.e.* une fourche consciente infinie) soit une branche infinie (*i.e.* une chaîne de pointeurs infinie). \square

Comme on le verra en Section 4.3, cette généralisation n'a d'intérêt que pour les arènes infinies, ou au moins de largeur infinie comme l'arène des entiers plats. En effet, on pourra prouver qu'une partie visible ou ultimement visible dans une arène finie est bornée si et seulement si sa relation \prec est bien fondée. Dans le cas des arènes de profondeur bien fondée, cette équivalence permettra de ramener les parties infinies à l'existence d'une fourche consciente infinie : il est évident que de tels phénomènes sont à bannir si on veut préserver la finitude des parties. Ces fourches conscientes seront au cœur de l'étude des *stratégies naïthériennes*, la plus grande classe de stratégie sur laquelle les résultats de ce chapitre s'appliquent.

Fourche conscientes opposées. À propos des fourches conscientes infinies, on serait tentés de proposer la conjecture suivante : si φ est une structure de pointeurs ultimement visible, alors toutes les fourches conscientes infinies ont la même polarité dans φ . Autrement dit, il est impossible que Joueur et Opposant aient chacun une fourche consciente infinie. Cette proposition paraît raisonnable, car chaque nouveau coup de la fourche pour Joueur «remet les vues à zéro» pour Opposant, et vice-versa. Elle s'avère finalement fautive, la Figure 4.5 représente deux fourches conscientes infinies opposées. On notera toutefois sur ce contre-exemple la présence très visible d'une chaîne de pointeurs infinie qui se charge de faire des aller-retours entre l'environnement de la fourche Joueur, et celui de la fourche Opposant. On verra en Section 4.3 qu'interdire une telle chaîne de pointeurs suffit à interdire l'existence de fourches conscientes infinies opposées : il y a donc *unicité du coupable*.

4.2 Théorèmes de finitude

Dans cette section, on va se baser sur les notions développées précédemment. Tout d'abord, on va énoncer les théorèmes de finitude, résultats centraux de ce chapitre. On en donnera ensuite plusieurs preuves très différentes, dont chacune mettra en exergue un aspect différent des structures de pointeurs : la première preuve les considérera comme des traces d'exécution de la machine à pointeurs, la seconde (plus sémantique) les verra comme des interactions entre objets abstraits appelés les *agents*, qui sont à la fois une restriction et une généralisation des stratégies. Finalement, la troisième preuve présentée aura un aspect plus combinatoire.

On présente maintenant les deux variantes du théorème de finitude. Ces deux variantes correspondent exactement aux paires de conditions présentées à la fin de la section précédente.

Théorème 4.2.1 (Finitude faible). *Soit φ une structure de pointeurs ultimement visible bornée, alors φ est finie.*

Démonstration. Trois preuves indépendantes, en Sections 4.2.1, 4.2.2 et 4.2.3. □

Ce théorème suffira au corollaire principal de ce chapitre, c'est-à-dire que les stratégies innocentes totales finies sont stables par composition. Il suffit aussi pour gérer le cas des stratégies dites *bornées* ; ces stratégies qui peuvent avoir une fonction de vue infinie, mais admettent une borne sur la taille de leur vue. Cependant, considérons la stratégie agissant sur l'arène $\text{nat} \rightarrow \text{com} \rightarrow \text{com}$, obtenue par l'interprétation du terme suivant :

```
let rec repeat n c =
  let z = n in
  if z = 0 then () else (c; repeat (z-1) c)
```

Aucun entier ne convient pour borner la taille de ses vues. En effet, si son argument est l'entier n , elle va exécuter n fois la commande c , et a donc au moins une vue plus grande que n pour tout entier n . Cette stratégie appartiendra cependant à une classe de stratégies plus générale appelées *stratégies naïves*. On aura pour ces stratégies aussi un résultat de préservation de la totalité par composition, mais pour le montrer on aura besoin de la version plus forte du théorème de finitude suivante :

Théorème 4.2.2 (Finitude forte). *Soit φ une structure de pointeurs ultimement visible telle que \prec_φ est bien fondée. Alors φ est finie.*

Démonstration. Section 4.2.3. □

4.2.1 PAM inversée et réduction linéaire de tête

Dans cette première partie, on va montrer comment on peut se servir de la machine à pointeurs développée dans le chapitre précédent pour démontrer le théorème de finitude faible. L'idée est qu'à chaque structure de pointeurs φ on va associer deux arbres de Böhm unaires U et V , telle que φ correspond à la séquence (p_i) d'une exécution de la PAM sur UV . L'idée est alors que par correction de la PAM, φ décrit en fait une chaîne de réduction linéaire de tête dirigée de UV , et puisque UV est bien typé φ doit être finie.

Définition 4.2.1. *Soit U un arbre de Böhm unaire, comme défini en Section 3.3.3. Si T est un sous-terme de U , on note $\text{var}(T)$ l'ensemble des variables du contexte de typage de T , avec l'addition de la variable éventuellement liée à la racine de T . Soit $x \in \text{var}(T)$ de type au moins 2. L'expansion de U à T par x , notée $\text{exp}(U, T, x)$, est définie par cas sur T :*

- Si $T = \lambda y.\mathfrak{X}$, on remplace T par $\lambda y.x(\lambda z.\mathfrak{X})$ (où z est une variable fraîche).
- Si $T = \lambda y.\bigoplus_{i \in I} h_i M_i$ avec l'un des h_i occurrence de x , on ne change rien.
- Sinon c'est qu'aucun h_i n'est occurrence de x , on remplace T par

$$\lambda y.\left(\bigoplus_{i \in I} h_i M_i\right) + (x\mathfrak{X})$$

Cette transformation préserve de type de l'arbre de Böhm, mais aussi le type de ses sous-termes.

Définition 4.2.2. Soit φ une structure de pointeurs visible de profondeur finie d . Un état de la PAM inversée sur φ comprendra les données suivantes :

- Deux arbres de Böhm unaires $\Gamma \vdash U_n : d + 2$ et $\Gamma \vdash V_n : d + 1$,
- Un état de PAM $(h_1, A_1, p_1), \dots, (h_n, A_n, p_n)$ sur $U_n V_n$ tel que pour tout $1 \leq i \leq n$, $p_i = \varphi(i)$. Les A_i doivent être des sous-termes en position d'argument, donc par propriété des arbres de Böhm unaires ils commencent tous par une abstraction.

Ces données doivent vérifier la condition de typage : le type de chaque A_i doit être $d - d_i + 1$, où s est la profondeur de φ et d_i la profondeur du coup i dans φ .

L'initialisation et les transitions de la PAM inversée seront très proches de celles de la PAM en situation de jeux, présentées à la fin du chapitre précédent. Simplement, au lieu de fixer les termes et d'en déduire la séquence de pointeurs, on fixe celle-ci et on en déduit de l'information sur les termes. D'un point de vue de sémantique, cela revient à récupérer U par le théorème de définissabilité à partir de la stratégie définie par les P -vues de φ , et V de la même façon à partir des O -vues de φ .

Initialisation de la RPAM. Soit φ une structure de pointeurs visible, l'initialisation de la RPAM sur φ est définie comme suit :

- $U_1 = \lambda x.x_0\mathfrak{X}_0$, $V_1 = (\lambda y.y_0\mathfrak{X}_1)$.
- $h_1 = x_0$, $A_1 = (\lambda y.y_0\mathfrak{X}_1)$ et $p_1 = 0$.

Transition de la RPAM. Supposons qu'on ait un état U_n, V_n et $(h_1, A_1, p_1), \dots, (h_n, A_n, p_n)$ de la RPAM sur une structure visible φ . On obtient l'état à $n + 1$ par les opérations suivantes :

1. Si $\varphi(n + 1) = \perp$, la machine s'arrête.
2. Sinon, on considère $A_{\varphi(n+1)}$, soit x la variable qui y est abstraite. Deux cas possibles :
 - a. Si il existe une occurrence x_i de x dans $\text{hoc}(A_n)$, on pose $h_{n+1} = x_i$, $A_{n+1} = \text{arg}(h_{\varphi(n+1)})$ et $p_{n+1} = \varphi(n + 1)$. U et V sont inchangés. Ce nouvel état vérifie bien la contrainte de typage, puisque $h_{\varphi(n+1)}$ est lié par $A_{\varphi^2(n+1)}$, lequel est de type $d - d_{\varphi^2(n+1)} + 1$ par la contrainte de typage. On en déduit que l'argument A_{n+1} de $h_{\varphi(n+1)}$ est de type $d - d_{\varphi^2(n+1)} + 1 - 2$, ce qui est bien égal à $d - d_{n+1} + 1$.
 - b. Sinon, supposons (sans perte de généralité) que A_n est dans U_n . On pose $U_{n+1} = \text{exp}(U_n, A_n, x)$ et $V_{n+1} = V_n$. On adapte également en conséquence les valeurs des A_i ⁴. Ainsi, $(h_1, A_1, p_1), \dots, (h_n, A_n, p_n)$ reste un état correct de la PAM sur $U_{n+1} V_{n+1}$. Le cas (a) s'applique alors sans problème.

⁴On peut supposer que les A_i sont donnés par des pointeurs vers le sous-terme correspondant de $U_n V_n$, de sorte qu'il n'est en fait pas nécessaire d'y faire l'expansion.

Terminaison. Le point clé de la démonstration est que maintenant, on peut donner plusieurs conditions équivalentes à l'arrêt de cette procédure, de sorte qu'elles doivent être équivalente entre elles. La première est évidente, il s'agit de la finitude de φ . En effet, on progresse d'une étape à chaque fois dans \mathcal{D}_φ et on s'arrête dès qu'on trouve $i \in \mathbb{N}$ tel que $\varphi(i) = \perp$. Voici la seconde condition :

Proposition 4.2.3. *Soit φ une structure de pointeurs visible, la RPAM sur φ s'arrête si et seulement si la procédure d'expansion n'est plus utilisée à partir d'un certain rang, c'est-à-dire qu'il existe N au delà duquel le cas (2b) n'est plus utilisé.*

Démonstration. Si (2b) n'est plus utilisé, on a $U_n = U_N$ et $V_n = V_N$ pour tout $n \geq N$. Par conséquent, on entretient une exécution de la PAM sur $U_N V_N$, et par correction de la PAM (Théorème 3.3.4), cette exécution décrit une chaîne de réduction linéaire de tête de $U_N V_N$. Mais la réduction linéaire de tête normalise par la Proposition 3.2.2, donc la machine s'arrête. Réciproquement, si la machine s'arrête le processus d'expansion s'arrête aussi. \square

Lemme 4.2.4. *La transition (2b) est utilisée un nombre fini de fois si et seulement si φ est bornée (i.e. a des vues de taille bornée).*

Démonstration. Commençons par montrer le sens direct, par contraposée : supposons donc que φ est non bornée. On montre en fait qu'à chaque étape i où on rencontre une taille de vue jamais obtenue auparavant, on utilise la transition (2b). C'est simplement par le Lemme 3.3.2 : si h_i est une occurrence de tête, l'ensemble des A_j pour $j \in \lceil \varphi \rceil(i) \wedge j \neq i$ est la chaîne de sur-termes de h_i . Si la vue atteint une longueur jamais atteinte auparavant, c'est donc que la chaîne de sur-termes atteint une longueur jamais atteinte, donc qu'on vient d'expanser le terme.

Montrons maintenant l'implication réciproque, encore une fois par contraposée. Supposons que la transition (2b) est utilisée un nombre infini de fois, supposons que c'est à des indices i_0, i_1, \dots . Par définition de cette transition, on a alors une chaîne infinie d'occurrences h_{i_0}, h_{i_1}, \dots . Pour chacune d'entre elles, la vue correspond à la chaîne de sur-termes par le Lemme 3.3.2. Mais nécessairement, les h_{i_k} admettent des chaînes de sur-termes non bornées puisqu'elles sont toutes différentes et que dans un arbre de Böhm unaire, les uniques branchements (i.e. les sommes non déterministes) ont une arité finie. Par conséquent, la suite $|\lceil \varphi \rceil(i_0)|, |\lceil \varphi \rceil(i_1)|, \dots$ est non bornée et le résultat est démontré. \square

Corollaire 4.2.5 (Finitude faible visible). *Soit φ une structure de pointeurs visible bornée, alors φ est finie.*

Démonstration. Si φ est bornée et visible, la RPAM n'utilise qu'un nombre fini de fois la transition (2b) par le Lemme 4.2.4. Par conséquent, φ est finie d'après la Proposition 4.2.3. \square

On vient donc de donner la première preuve du théorème de finitude. On peut remarquer plusieurs choses à son sujet. Tout d'abord, le fonctionnement de la preuve fait qu'elle se restreint au cas des structures de pointeurs *visibles*, et non pas *ultimement visibles*. Ce n'est pas un gros problème, parce que les théorèmes de préservation de la totalité par composition peuvent se faire avec le cas visible (même si cela demande un peu de technique en plus), et que de toute façon les prochaines preuves démontreront le cas plus général.

Il peut être intéressant de remarquer que la machine inversée fait plus que permettre de démontrer le théorème de finitude faible, elle établit en fait une équivalence très forte entre

le théorème de finitude faible et la terminaison de la machine à pointeurs sur les situations de jeux, et donc de la terminaison de la réduction linéaire de tête pour les termes typés. Il est notable que cette preuve d'équivalence se fait sans induction, ce qui souligne la connection très précise entre les deux.

On peut quelque peu regretter la dépendance de cette preuve aux arguments syntaxiques de normalisation. Le but de ces théorèmes de finitude étant de donner des arguments sémantiques pour la préservation de la totalité par composition, on aimerait en avoir des preuves plus légères, en particulier sans recours à la syntaxe. Ainsi, la prochaine section développera une preuve plus sémantique, par réalisabilité, qui est en quelque sorte le résultat de l'élimination des coupures sur la preuve indirecte présentée au dessus : on remet en scène l'argument de réalisabilité de la Section 3.1, mais cette fois directement dans le langage des structures d'interaction.

4.2.2 Agents, interactions et réalisabilité

Depuis le début de cette partie, on a défendu la thèse que dans un calcul, les propriétés de normalisation ne dépendaient que de la profondeur des types, et pas de leur structure. Cette vision des choses a conduit à de nombreuses simplifications : le passage des types simples au cas unaire, puis l'effondrement qui a permis d'oublier des arènes et la restriction aux structures de pointeurs. Dans cette section, on poursuit ce processus de simplification en oubliant cette fois la structure des termes et des stratégies en ne gardant que des bornes sur la longueur de leurs vues. On pourra alors voir les structures de pointeurs directement comme des interactions entre ces *agents* simplifiés, et en tirer une nouvelle preuve, plus directe, du théorème de finitude.

Une étape de réduction linéaire de tête correspond à une descente d'un cran sur une structure de pointeurs. Cela nous amènera naturellement à considérer comme objet primitif non pas les structures de pointeurs en elles mêmes, mais les couples (φ, i) avec φ une structure de pointeurs et $i \in \mathcal{D}_\varphi$ un point de φ désigné comme *origine*. Tout ce qui se passe précédemment est considéré comme faisant partie de l'*histoire* de l'exécution. C'est la contrepartie sémantique du fait que si on examine une réduction linéaire de tête entre deux termes clos $M = \lambda x.xM'$ et N , on va observer la réduction :

$$(\lambda x.xM')N \rightsquigarrow^{LHR} (\lambda x.NM')N$$

Donc, l'interaction entre les deux termes clos M et N se résoud sur l'interaction entre N et M' — mais M' n'est plus clos. Là où l'interaction close entre M et N était interprétée par une structure φ , l'interaction ouverte entre N et M' sera représentée par le couple $(\varphi, 1)$. De la même façon, si les termes clos sont représentés par des entiers, on donnera une définition d'*agent* pour représenter les *fermetures*, comme $M^{!x \mapsto N}$ dans l'exemple ci-dessus.

On va maintenant définir dans notre cadre sémantique une série de notions inspirées du calcul, notions qui permettront d'écrire la preuve par réalisabilité du théorème de finitude.

Types. Pas de surprise ici, les types seront toujours modélisés par des entiers. Intuitivement, un tel entier représentera la profondeur de l'arène ambiante.

Termes clos. Comme annoncé plus haut, les termes clos seront représentés par des entiers. Deux entiers n et p vont admettre comme interactions toutes les structures de pointeurs dont les longueurs de *P-vues/O-vues* sont compatibles avec n et p .

Définition 4.2.3. Une interaction fermée entre deux entiers n et p est une structure de pointeurs visible φ telles que :

$$\forall i \in \mathcal{D}_\varphi, |\ulcorner\varphi^\top(i)| \leq n \wedge |\llcorner\varphi_\perp(i)| \leq p + 1$$

L'ensemble de toutes les interactions fermées entre n et p sera noté $n \star p$.

Remarque. On a ici une différence de traitement entre les P -vues et les O -vues. La taille des P -vues est limitée à n , alors que la taille des O -vues est limitée à $p + 1$. Pour expliquer cela, il faut imaginer que la partie se joue sur une arène de la forme $A \Rightarrow o$. Le premier coup est dans l'occurrence de o tout à droite, alors que le reste de la partie est sur A . L'ensemble des P -vues encode alors une stratégie sur $A \Rightarrow o$, alors que l'ensemble des O -vues correspond plutôt à une partie sur A . Pour évaluer les longueurs des O -vues, il faut donc éviter de tenir compte du premier coup sur o , d'où la borne en $p + 1$.

Contextes. Comment représenter une interaction entre deux termes ouverts dans ce cadre ? Comme mentionné auparavant, ce sera fait par l'intermédiaire de couples (φ, i) entre une structure de pointeurs φ et une origine $i \in \mathcal{D}_\varphi$. Si on a un coup $j \geq i$ let que $\varphi(j) < i$, c'est que l'agent jouant à j fait un *appel à l'environnement*.

Termes généralisés. Il est désormais nécessaire d'étendre notre notion de terme à cette forme plus générale d'interaction. Considérons une structure de pointeurs φ et une origine $i \in \mathcal{D}_\varphi$. Quels sont les termes dont l'interaction commence à i ? Pour répondre à cette question, il faut commencer par calculer la taille des vues atterrissant sur i . On pourrait les obtenir comme les tailles maximales de vues dans la sous-partie des coups justifiés héréditairement par i , mais cela exigerait de définir préalablement la restriction d'une partie aux successeurs de i , ce qui s'avère trop lourd techniquement. De manière plus élémentaire, on les calcule comme suit :

Définition 4.2.4. Soit φ une structure de pointeurs, et soit $i \in \mathcal{D}_\varphi$. La vue maximale depuis i , notée $\overline{\ulcorner\varphi_\perp}(i)$, est définie comme suit :

$$\overline{\ulcorner\varphi_\perp}(i) = \max \left(\begin{array}{l} \max_{\substack{i \in \ulcorner\varphi_\perp(j) \\ i \equiv_2 j}} |\ulcorner\varphi_\perp(j)| - |\ulcorner\varphi_\perp(i)| + 1, \\ \max_{\substack{i \in \llcorner\varphi^\top(j) \\ i \not\equiv_2 j}} |\llcorner\varphi^\top(j)| - |\ulcorner\varphi_\perp(i)| + 1 \end{array} \right)$$

De même, on définit la covue maximale depuis i par :

$$\overline{\llcorner\varphi^\top}(i) = \max \left(\begin{array}{l} \max_{\substack{i \in \llcorner\varphi^\top(j) \\ i \equiv_2 j}} |\llcorner\varphi^\top(j)| - |\llcorner\varphi^\top(i)| + 1, \\ \max_{\substack{i \in \ulcorner\varphi_\perp(j) \\ i \not\equiv_2 j}} |\ulcorner\varphi_\perp(j)| - |\llcorner\varphi^\top(i)| + 1 \end{array} \right)$$

Fermetures. La définition d'au dessus ne décrit que partiellement les agents interagissant dans un contexte ouvert. Considérons une fermeture (au sens de la Section 3.3.2) M^σ . La définition précédente décrit (la taille de l'arbre de Böhm de) M , mais la fermeture doit spécifier

aussi le comportement de l'environnement lorsque M fait appel à ses variables libres. Dans notre cadre, cela signifie que nos agents doivent porter de l'information sur le comportement de leur Opposant lorsqu'elles pointent sur un coup antérieur à l'origine de la structure.

Définition 4.2.5. *Un agent est la donnée d'un entier n et d'un k -uplet d'agents a_1, \dots, a_k , noté $n[a_1, \dots, a_k]$. La définition est bien fondée car on se retrouve finalement confronté au cas où $k = 0$. L'ensemble des agents sera noté \mathcal{A} .*

Remarque. Une explication alternative de cette définition est qu'elle permet de parler de stratégies qui ne sont pas en forme β -normale. En effet, un agent $n[a_1, \dots, a_k]$ peut être compris comme un terme n avec k variables libres, et avec des substitutions a_1, \dots, a_k «en attente». C'est d'ailleurs cette intuition qui justifie la notation $n[a_1, \dots, a_k]$, inspirée de la notation courante pour la substitution. Peut-être serait-il possible de s'inspirer de ces définitions pour développer une sémantique des jeux où on sépare le mécanisme d'interaction parallèle et celui de masquage nécessaires à la composition : la construction de l'agent $n[a_1, \dots, a_k]$ serait alors obtenue par interaction parallèle et le masquage correspondrait à la renormalisation vers un arbre de Böhm. Malheureusement, ces idées devront attendre des travaux ultérieurs.

Il reste à définir comment ces agents interagissent. L'idée est naturelle, même si la mise en œuvre est quelque peu technique : le résultat d'une interaction entre deux agents a et b est un couple (φ, i) où les vues à partir de i sont compatibles avec les tailles de a et b , et tel que les appels au contexte sont compatibles avec les parties «substitutions» de a et b .

Définition 4.2.6. *Soit $a = n[a_1, \dots, a_p]$ un agent. L'ensemble des traces de a , noté $Tr(a)$, est l'ensemble des couples (φ, i) , avec $i \in \mathcal{D}_\varphi$, tels que :*

- (1) $\overline{\varphi}(i) \leq n$
- (2) $|\{j \in \overline{\varphi}^{-1}(i) \mid (j \equiv_2 i) \wedge (j < i)\}| = p$, notons $\{j_1, \dots, j_p\}$ cet ensemble, réordonné par $<$.
- (3) Pour tout $m \in \{1, \dots, p\}$, pour tout $x \in \varphi^{-1}(j_m)$ tel que $x \geq i$, $(\varphi, x) \in Tr(a_m)$.
- (4) φ est visible après i .

La première condition exprime le fait que la taille des vues à i ne dépasse pas n . La seconde, quant à elle, contraint les structures à fournir un contexte exactement de la même taille que l'environnement (*i.e.* la liste a_1, \dots, a_p) de a . Finalement, on impose que les appels à ces éléments du contexte respectent bien les agents de l'environnement de a .

Définition 4.2.7. *Si a et b sont des agents, on définit l'ensemble $a \star b$ des interactions entre a et b comme suit :*

$$a \star b = \{(\varphi, i) \in Ptr \times \mathbb{N} \mid (\varphi, i) \in Tr(a), \forall x \in \varphi^{-1}(i), (\varphi, x) \in Tr(b)\}$$

Le lecteur notera la surcharge de la notation \star . C'est justifié parce que les deux notions coïncident sur les interactions closes, comme démontré par le lemme suivant.

Lemme 4.2.6. *Soient $n, p \in \mathbb{N}$, alors*

$$\varphi \in n \star p \Leftrightarrow (\varphi, 0) \in n[] \star p[]$$

Démonstration. Commençons par remarquer que $\overline{\varphi_{\downarrow}}(0) = \max_{i \in \mathcal{D}_{\varphi}} |\overline{\varphi^{\uparrow}}(i)|$:

$$\begin{aligned} \overline{\varphi_{\downarrow}}(0) &= \max \left(\max_{j \equiv_2 0} |\overline{\varphi_{\downarrow}}(j)| - |\overline{\varphi_{\downarrow}}(0)| + 1, \max_{j \not\equiv_2 0} |\underline{\varphi^{\uparrow}}(j)| - |\overline{\varphi_{\downarrow}}(0)| + 1 \right) \\ &= \max \left(\max_{j \equiv_2 0} |\overline{\varphi_{\downarrow}}(j)|, \max_{j \not\equiv_2 0} |\underline{\varphi^{\uparrow}}(j)| \right) \\ &= \max_{j \in \mathcal{D}_{\varphi}} |\overline{\varphi^{\uparrow}}(j)| \end{aligned}$$

De façon symétrique, on a $\underline{\varphi^{\uparrow}}(0) = \max_{i \in \mathcal{D}_{\varphi}} |\underline{\varphi_{\downarrow}}(i)|$.

$$\begin{aligned} \underline{\varphi^{\uparrow}}(0) &= \max \left(\max_{j \equiv_2 0} |\underline{\varphi^{\uparrow}}(j)| - |\underline{\varphi^{\uparrow}}(0)| + 1, \max_{j \not\equiv_2 0} |\overline{\varphi_{\downarrow}}(j)| - |\underline{\varphi^{\uparrow}}(0)| + 1 \right) \\ &= \max \left(\max_{j \equiv_2 0} |\underline{\varphi^{\uparrow}}(j)|, \max_{j \not\equiv_2 0} |\overline{\varphi_{\downarrow}}(j)| \right) \\ &= \max_{j \in \mathcal{D}_{\varphi}} |\underline{\varphi_{\downarrow}}(j)| \end{aligned}$$

Ces deux faits suffisent à montrer que les contraintes de taille sont les mêmes des deux côtés. Les autres conditions sont triviales : φ est supposée visible à partir de 0 des deux côtés, et les conditions (2) et (3) de la définition de $n[] \star p[]$ sont triviales puisque $\{j \in \underline{\varphi^{\uparrow}}(0) \mid (j \equiv_2 0) \wedge (j < 0)\} = \emptyset$. \square

Recomposition. Si φ est visible et bornée par M , il est évident que φ est une trace d'interaction entre agents. Plus précisément, on a $\varphi \in M \star M + 1$ (ou, de façon équivalente, que $(\varphi, 0) \in M[] \star M + 1[]$. Voir le Lemme 4.2.6). Si φ n'est que ultimement visible, c'est moins évident mais cependant assuré par le lemme suivant.

Lemme 4.2.7. *Soit φ une structure de pointeurs, dont les vues sont bornées par $M \in \mathbb{N}$, soit i l'indice de visibilité de φ . Alors :*

- (1) *On peut retrouver un agent $a_i \in \mathcal{A}$ tel que $(\varphi, i) \in Tr(a_i)$. De plus, l'agent obtenu ne dépend que de $\varphi(i)$.*
- (2) *On peut retrouver un agent $b_i \in \mathcal{A}$ tel que $(\varphi, i) \in a_i \star b_i$.*

Démonstration. (i). Par induction sur $\varphi(i)$. Si $\varphi(i) = 0$, alors $(\varphi, i) \in Tr(M)$. Sinon on définit :

$$\{j_1, \dots, j_p\} = \{x \in \underline{\varphi^{\uparrow}}(i) \mid x \equiv_2 i \wedge x < i\}.$$

En effet, $\{x \in \underline{\varphi^{\uparrow}}(i) \mid x \equiv_2 i \wedge x < i\}$ ne peut pas être vide si $\varphi(i) \neq 0$, puisqu'il contient au moins $\varphi(i) - 1$. Maintenant, pour chacun des j_x et pour tout $y > i \in \mathcal{D}_{\varphi}$ tel que $\varphi(y) = j_x$, on a par hypothèse d'induction un agent a_x tel que $(\varphi, y) \in Tr(a_x)$. De plus, l'hypothèse d'induction assure aussi que a_x ne dépend pas du choix de y . Par conséquent :

$$(\varphi, i) \in Tr(M[a_1, \dots, a_p]).$$

Il reste à montrer que cet agent ne dépend pas du choix de i mais uniquement de $\varphi(i)$. Il suffit de remarquer que si $i' \in \mathcal{D}_{\varphi}$ est tel que φ est visible après i' et avec $\varphi(i) = \varphi(i')$, on aura aussi :

$$\{x \in \underline{\varphi^{\uparrow}}(i) \mid x \equiv_2 i \wedge x < i\} = \{x \in \underline{\varphi^{\uparrow}}(i') \mid x \equiv_2 i' \wedge x < i'\}.$$

On pourra donc appliquer à i' exactement le même argument qu'à i , et on obtiendra aussi $(\varphi, i') \in Tr(M[a_1, \dots, a_p])$.

(ii). Soit $i = 0$, auquel cas $(\varphi, i) \in M \star M$. Sinon, on commence par récupérer par (i) l'agent a_i tel que $(\varphi, i) \in Tr(a_i)$. Ensuite, on calcule :

$$\{j_1, \dots, j_q\} = \{x \in \lceil \varphi \rceil(i) \mid x \neq_2 i\}.$$

Pour chaque j_x , (i) fournit un agent b_x tel que, pour tout $y > i$ tel que $\varphi(y) = j_x$, $(\varphi, y) \in Tr(b_x)$, et cet agent ne dépend pas du choix de y . On a donc bien $(\varphi, i) \in a_i \star M[b_1, \dots, b_q]$. \square

En fait, le fait d'être une trace d'interaction entre agents *caractérise* les structures de pointeurs bornées ultimement visibles : on peut montrer facilement la réciproque du lemme ci-dessus, c'est-à-dire que si φ est ultimement visible et que $(\varphi, i) \in a \star b$, alors φ est bornée.

Réduction des agents. Puisque les agents ont été inspirés par les λ -termes, il est naturel de voir si on peut les équiper à leur tour d'une dynamique, indépendante de celle des structures de pointeurs. C'est cette idée que nous allons investiguer ici. On va voir qu'effectivement ils disposent d'une réduction naturelle, mais qu'en fait celle-ci sous-tend la dynamique des structures de pointeurs.

Remarque. Dans l'article [CH09], on présente à ce point une définition de réalisabilité ainsi qu'une preuve directe d'un lemme d'adéquation correspondant, permettant de démontrer le théorème de finitude. Dans cette thèse, on va développer un peu plus et décomposer cette preuve par réalisabilité en deux étapes. D'abord, on va exhiber une réduction sur les couples d'agents compatible avec le «déroutement»⁵ d'une structure de pointeurs, puis montrer par réalisabilité que cette réduction termine.

Définition 4.2.8. Si $n[a_1, \dots, a_p]$ et b sont des agents, on définit la relation de réduction non déterministe suivante :

$$\begin{aligned} (n[a_1, \dots, a_p], b) &\rightarrow (b, (n-2)[a_1, \dots, a_p, b]) \\ (n[a_1, \dots, a_p], b) &\rightarrow (a_i, (n-2)[a_1, \dots, a_p, b]) \end{aligned}$$

Cette réduction s'arrête lorsque $n \leq 2$.

Cette transformation décrit à la perfection le passage de l'étape (φ, i) à $(\varphi, i+1)$ sur une structure de pointeurs, comme le montre le lemme suivant :

Lemme 4.2.8 (Simulation directe). Soit $(\varphi, i) \in a \star b$, alors si $\varphi(i+1) \neq \perp$, il existe des agents a', b' tels que $(a, b) \rightarrow (a', b')$ et $(\varphi, i+1) \in a' \star b'$.

Démonstration. Supposons φ toujours définie en $i+1$, et posons $a = n[a_1, \dots, a_p]$. Notons que dans tous les cas, il va falloir montrer que $(\varphi, i+1) \in c \star (n-2)[a_1, \dots, a_p, b]$, avec c valant b ou l'un des a_i . Pour montrer ce résultat, plusieurs étapes sont nécessaires :

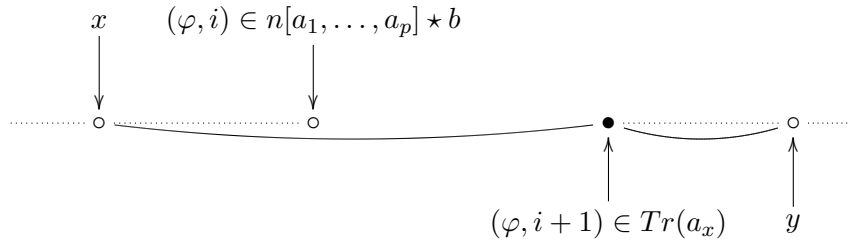
1. Exhiber l'agent c correspondant et montrer que $(\varphi, i+1) \in Tr(c)$;
2. Montrer que pour tout $y \in \varphi^{-1}(i+1)$, $(\varphi, y) \in Tr((n-2)[a_1, \dots, a_p, b])$.

⁵Par «déroutement», on veut dire le fait de passer de (φ, i) à $(\varphi, i+1)$.

1. Pour trouver c , il faut examiner la valeur de $\varphi(i+1)$. Intuitivement, si φ pointe vers le coup précédent, il fait un appel à son argument b , lequel va arriver en tête. S'il pointe sur un coup précédent il fait alors un appel au contexte et c'est l'un des a_i qui va arriver en tête. Plus formellement, soit $\{j_1, \dots, j_{p+1}\} = \{j \in \ulcorner \varphi \urcorner(i+1) \mid j \not\equiv_2 i+1\}$. Le cardinal de cet ensemble est bien $p+1$, puisqu'on avait $(\varphi, i) \in n[a_1, \dots, a_p] \star b$, donc $|\{j \in \ulcorner \varphi \urcorner(i) \mid j < i \wedge j \equiv_2 i\}| = p$ et $\{j \in \ulcorner \varphi \urcorner(i+1) \mid j \not\equiv_2 i+1\} = \{i\} \cup \{j \in \ulcorner \varphi \urcorner(i) \mid j < i \wedge j \equiv_2 i\}$. Par visibilité de φ après i , il existe $x \in \{1, \dots, p+1\}$ tel que $\varphi(i+1) = j_x$. Puisque $(\varphi, i) \in n[a_1, \dots, a_p] \star b$, alors :

- Soit $x \leq p$ et $(\varphi, i+1) \in Tr(a_x)$: dans ce cas on pose $c = a_x$.
- Soit $x = p+1$ et $(\varphi, i+1) \in Tr(b)$; dans ce cas on pose $c = b$.

2. Il nous faut maintenant montrer que pour tout $y \in \varphi^{-1}(i+1)$, $(\varphi, y) \in Tr((n-2)[a_1, \dots, a_p, b])$. Soit donc un tel $y \in \varphi^{-1}(i+1)$. Illustrons l'état courant de φ :



Le premier point à vérifier est que $\overline{\ulcorner \varphi \urcorner}(y) \leq n-2$, mais c'est clair puisque par définition $\overline{\ulcorner \varphi \urcorner}(i) \geq \overline{\ulcorner \varphi \urcorner}(y) + 2$. Montrons maintenant la propriété (2) de la définition de Tr :

$$\begin{aligned} \ulcorner \varphi \urcorner(y) &= \{y\} \cup \ulcorner \varphi \urcorner(i+1) \\ &= \{y, i+1\} \cup \ulcorner \varphi \urcorner(i) \end{aligned}$$

Donc $\{j \in \ulcorner \varphi \urcorner(y) \mid (j < y) \wedge (j \equiv_2 y)\} = \{j_1, \dots, j_p, i\}$, de cardinal $p+1$ comme nécessaire. La propriété (3) découle immédiatement du fait que $(\varphi, i) \in n[a_1, \dots, a_p] \star b$. La (4) est évidente, puisque φ était déjà supposée visible après i . On a fini de montrer que $(\varphi, i+1) \in c \star (n-2)[a_1, \dots, a_p, b]$. \square

Par itération de ce lemme, on peut associer à chaque structure de pointeurs visible et bornée φ des agents a, b tels que $(\varphi, 0) \in a \star b$ et une séquence de réduction aussi longue que φ :

$$(a, b) \rightarrow (a_1, b_1) \rightarrow (a_2, b_2) \rightarrow \dots$$

Le lemme suivant démontre l'autre sens de la simulation. Il n'est pas nécessaire pour le théorème de normalisation en lui-même, mais semble le bienvenu ici en ce sens qu'il achève de montrer que les structures de pointeurs visibles correspond exactement aux séquences de la réduction \rightarrow .

Lemme 4.2.9 (Simulation inverse). *Soit $(a_0, b_0) \rightarrow \dots \rightarrow (a_{i+1}, b_{i+1})$ une chaîne de réduction et φ une structure de pointeurs visible telle que pour tout $0 \leq j \leq i$, on ait $(\varphi, j) \in a_j \star b_j$. Alors il existe une structure visible φ' définie au moins jusqu'à $i+1$ coïncidant avec φ sur $0, \dots, i$ et telle que pour tout $0 \leq j \leq i+1$, on ait $(\varphi', j) \in a_j \star b_j$.*

Démonstration. Posons $(a_i, b_i) = (n_i[a_i^1, \dots, a_i^p], b_i)$, et raisonnons sur les deux cas possibles de la réduction $(a_i, b_i) \rightarrow (a_{i+1}, b_{i+1})$.

- Si $(a_{i+1}, b_{i+1}) = (b_i, (n_i - 2)[a_i^1, \dots, a_i^p, b_i])$, alors on définit φ' par :

$$\begin{aligned}\varphi'(j) &= \varphi(i) & 0 \leq j \leq i \\ \varphi'(i+1) &= i \\ \varphi'(j) &= \perp & \text{si } j > i+1.\end{aligned}$$

- Si $(a_{i+1}, b_{i+1}) = (a_i^j, (n_i - 2)[a_i^1, \dots, a_i^p, b_i])$, alors comme $(\varphi, i) \in Tr(n_i[a_i^1, \dots, a_i^p])$, en particulier $\varphi^{-1}(i)$ comprend p éléments $\{x_1, \dots, x_p\}$ de même polarité que i , où $\varphi'(i+1)$ peut potentiellement pointer en préservant la visibilité. La réduction $(a_i, b_i) \rightarrow (a_i^j, (n_i - 2)[a_i^1, \dots, a_i^p, b_i])$ place a_i^j en tête, ainsi on pose

$$\begin{aligned}\varphi'(j) &= \varphi(i) & 0 \leq j \leq i \\ \varphi'(i+1) &= x_j \\ \varphi'(j) &= \perp & \text{si } j > i+1.\end{aligned}$$

Par construction, ces structures de pointeurs sont toujours visibles. Il faut alors montrer que pour tout $0 \leq j \leq i+1$, on a $(\varphi', j) \in a_j \star b_j$. Cette vérification se décompose en deux parties. D'une part, il faut que pour tout $0 \leq j \leq i+1$, le nombre d'éléments dans $\varphi^{-1}(j)$ de même polarité que j corresponde bien au nombre de sous-arbres de la racine de a_j . Mais pour $0 \leq j \leq i$ c'est trivial, puisque pour ces indices φ' coïncide avec φ . Pour $i+1$ ce n'est guère plus difficile, puisqu'on a choisi $\varphi'(i+1)$ exactement pour ça. Il faut alors montrer que pour tout $0 \leq j \leq i+1$, la taille des vues arrivant sur j est bien bornée par la racine de a_j . Pour toutes les vues ne passant pas par $i+1$ c'est trivial par construction de φ' , sinon c'est assuré par la validité de la réduction $(a_i, b_i) \rightarrow (b_i, (n_i - 2)[a_i^1, \dots, a_i^p, b_i])$ qui sous-entend que $n_i > 2$, or les vues arrivant sur i n'auront pas une taille plus grande que 2. \square

On peut résumer les deux lemmes ci-dessus par la proposition précédente :

Proposition 4.2.10. *Si $a, b \in \mathcal{A}$, alors on a une bijection préservant la finitude (et la longueur dans le cas fini) entre les structures de pointeurs φ visibles telles que $(\varphi, 0) \in a \star b$ et les séquences de réduction $(a, b) \rightarrow (a_1, b_1) \rightarrow (a_2, b_2) \rightarrow \dots$*

Démonstration. Il suffit de voir qu'à agents (a, b) initialement fixes, les deux constructions ci-dessus sont inverses l'une de l'autre. \square

Ainsi une séquence de réduction d'agents est une autre façon de décrire une structure d'interaction. Cette nouvelle représentation dispose de quelques atouts, comme nous allons le voir maintenant.

Agents et arbres. On va maintenant oublier les structures de pointeurs et se concentrer uniquement sur les agents et leur réduction. Commençons par opérer quelques simplifications. Pour commencer, notons que toute paire d'agents peut être transformée de façon injective en un seul agent :

$$(n[a_1, \dots, a_p], b) \mapsto n[a_1, \dots, a_p, b]$$

Notons désormais $a \cdot b$ cette opération consistant à rajouter b comme dernier fils de la racine de a . Par cette injection, la réduction sur les couples d'agents se transporte de façon non ambiguë en une réduction sur les agents :

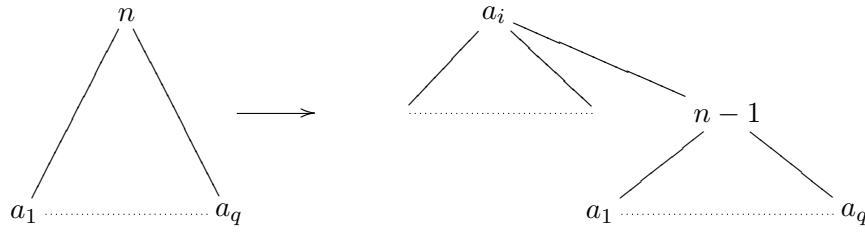


FIG. 4.6 – Règle de réécriture simplifiée sur les agents.

$$\begin{aligned} n[a_1, \dots, a_p, b] &\rightarrow b \cdot (n-2)[a_1, \dots, a_p, b] \\ n[a_1, \dots, a_p, b] &\rightarrow a_i \cdot (n-2)[a_1, \dots, a_p, b] \end{aligned}$$

On se rend alors compte que ces deux cas sont en réalité instances de cette seule et unique règle :

$$n[a_1, \dots, a_p] \rightarrow a_i \cdot (n-2)[a_1, \dots, a_p]$$

La dernière simplification consiste à remplacer le $n-2$ par un $n-1$ dans cette réduction, la condition d'arrêt $n \leq 2$ devient $n = 0$, et chaque entier k est remplacé par $\lfloor (k-1)/2 \rfloor$ (ou $\lfloor x \rfloor$ est la *partie entière* de x). Cette simplification n'est plus injective, mais il est immédiat de voir qu'elle ne change rien vis-à-vis de la réduction. Après toutes ces simplifications, convenons de voir les agents comme des arbres d'entiers : dans l'écriture $n[a_1, \dots, a_p]$, n décrit l'étiquette de la racine et chaque a_i décrit un sous-arbre. La réduction peut alors être résumée par la Figure 4.6. Remarquons que les transformations effectuées ne changent rien aux réductions possibles, ainsi on a préservé la Proposition 4.2.10 : chaque chaîne de cette réduction correspond très précisément à une structure de pointeurs, et réciproquement. On va maintenant étudier la terminaison de ce système.

Terminaison. Comme dans [CH09], ce sont des entiers qui vont ici jouer le rôle des types. On va interpréter chaque entier n par un ensemble d'arbres $|n|$, ses *réalisateurs*. Ces réalisateurs auront entre autres la propriété qu'ils seront fortement normalisants pour cette réduction. Ensuite, on montrera par induction sur les arbres un lemme d'*adéquation* assurant que tous les arbres sont dans $|n|$, pour n bien choisi.

Définition 4.2.9 (Orthogonalité). Soient $a, b \in \mathcal{A}$. On définit :

$$a \perp b \Leftrightarrow a \cdot b \text{ normalise fortement.}$$

Cette relation d'orthogonalité permet de définir l'orthogonal d'un ensemble $E \subset \mathcal{A}$:

$$E^\perp = \{x \in \mathcal{A} \mid \forall y \in E, x \perp y\}$$

Définition 4.2.10 (Réalisateurs). Pour chaque entier $n \in \mathbb{N}$, on définit par récurrence les réalisateurs de n :

$$\begin{aligned} |0| &= \{0[a_1, \dots, a_p] \mid a_1, \dots, a_p \in \mathcal{A}\} \\ |n+1| &= |n|^\perp \end{aligned}$$

Lemme 4.2.11 (Adéquation). *Si $d_1, \dots, d_p, d \geq 2n$, si pour tout $i \in \{1, \dots, p\}$ on a $a_i \in |d_i|$, alors $n[a_1, \dots, a_p] \in |d|$.*

Démonstration. Par récurrence sur n . Si $n = 0$, alors si $d = 0$ c'est vrai par définition. Sinon soit $b \in |d - 1|$. Il faut alors montrer que l'arbre $n[a_1, \dots, a_p, b]$ normalise fortement mais c'est trivial, puisqu'il n'y a pas de réduction possible. Supposons maintenant qu'on ait $n > 0$. Notons qu'on a alors $d \geq 2n \geq 2$, donc pour montrer que $a = n[a_1, \dots, a_p] \in |d|$ il faut montrer que $a \in |d - 1|^\perp$. Soit donc $b \in |d - 1|$, étudions les réductions possibles de $a \cdot b = n[a_1, \dots, a_p, b]$. On distingue deux cas, selon que c'est l'un des a_i ou b qui passe en tête.

- Si la réduction est $a \cdot b \rightarrow b \cdot (n - 1)[a_1, \dots, a_p, b]$. Pour tout i , on a alors $d_i \geq 2n \geq 2(n - 1)$, et $d - 2 \geq 2n - 2 \geq 2(n - 1)$ (donc $d - 1 \geq 2(n - 1)$), ainsi on sait par hypothèse de récurrence que $(n - 1)[a_1, \dots, a_p, b] \in |d - 2|$. Or $b \in |d - 1| = |d - 2|^\perp$ donc $b \cdot (n - 1)[a_1, \dots, a_p, b]$ est fortement normalisant.
- Si la réduction est $a \cdot b \rightarrow a_i \cdot (n - 1)[a_1, \dots, a_p, b]$. Pour tout i , on a alors $d_i \geq 2n \geq 2(n - 1)$, $d - 1 \geq 2n - 1 \geq 2(n - 1)$ et $d_i - 1 \geq 2n - 1 \geq 2(n - 1)$, ainsi on sait par hypothèse de récurrence que $(n - 1)[a_1, \dots, a_p, b] \in |d_i - 1|$. Mais $a_i \in |d_i| = |d_i - 1|^\perp$, donc $a_i \cdot (n - 1)[a_1, \dots, a_p, b]$ est fortement normalisant.

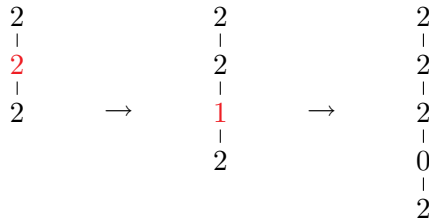
Ainsi tous les réduits de $a \cdot b$ sont fortement normalisants, donc $a \cdot b$ l'est lui-même. Donc $a \perp b$, et on a bien $a \in |d|$. □

Remarque. Ce lemme a été obtenu par adaptation du Lemme 3.1.1 suivi de nombreuses simplifications, ce qui explique que ses conditions semblent ad-hoc. De nombreuses variations ont été étudiées dans l'espoir de rendre cette preuve plus intuitive, sans succès. Quoi qu'il en soit, on peut en déduire immédiatement les deux résultats suivants :

Proposition 4.2.12 (Normalisation). *La réduction \rightarrow est fortement normalisante sur \mathcal{A} .*

Démonstration. Soit $a \in \mathcal{A}$. Prenons N l'entier maximum qui apparaisse dans a . Alors, par une induction immédiate et en utilisant le lemme d'adéquation à chaque étape, on montre que $a \in |2N + 1|$. De même, le lemme d'adéquation montre immédiatement que $0 \in |2N|$. Donc $a \cdot 0$ est fortement normalisant, mais toute réduction infinie de a serait également une réduction possible de $a \cdot 0$, donc a est fortement normalisant. □

Remarque. Le «fortement» du «fortement normalisante» de l'énoncé précédent concerne le choix non déterministe du réduit, pas le choix du redex qui est toujours celui en position de tête. La propriété de normalisation n'est plus vraie si on autorise la réduction en profondeur, donnons-en ici un contre-exemple. À chaque étape, on met en couleur la racine de l'arbre servant de redex.



Et la réduction boucle, en ignorant le sous-arbre commençant par 0.

Théorème 3 (Finitude faible). *Soit φ une structure de pointeurs ultimement visible bornée, alors φ est finie.*

Démonstration. Supposons que les vues de φ sont bornées par M , et que i est l'indice de visibilité de φ . Alors, par le Lemme 4.2.7, on peut reconstruire deux agents a et b tels que $(\varphi, i) \in a \star b$. Par la Proposition 4.2.12, on sait alors que le couple (a, b) est fortement normalisant pour la réduction \rightarrow , ainsi le Lemme 4.2.8 assure que φ est finie. \square

Remarque. Ainsi, on a réduit l'étude des structures de pointeurs bornées à l'étude d'une réduction simple sur les arbres d'entiers, donnant ainsi une nouvelle preuve du théorème de finitude : c'est un peu la dernière étape de la procédure de simplification commencé au début de cette partie, avec l'introduction du λ -calcul unaire. Cette réduction peu bien sur paraître d'intérêt un peu périphérique, cependant il convient de garder en mémoire que par la Proposition 4.2.10, elle n'est en fait rien d'autre qu'une façon alternative de voir les structures de pointeurs visibles. Ainsi, on espère que sa nature combinatoire puisse donner lieu à l'établissement d'une estimation sur les longueurs de réduction, estimation qui se porterait aussitôt sur les structures de pointeur, et via la PAM, sur la réduction linéaire de tête des termes typés.

4.2.3 Approche directe

Présentons finalement une dernière preuve, qui résoudra cette fois la version forte du théorème de finitude. Elle n'est pas nouvelle : elle apparaît pour la première fois dans [Coq95] sous le nom de «*no infinite chattering*», et dans un vocabulaire assez différent de celui présenté ici. On peut la trouver aussi dans [Cur98].

Le point de départ est une propriété des parties visibles qui semble être dans le folklore de la sémantique de jeux : elle est connue des spécialistes mais on n'en trouve pas de preuve écrite dans la littérature, du moins pas dans le vocabulaire de la sémantique de jeux (elle est prouvée dans [Coq95] et [Cur98]). Supposons qu'on ait une structure de pointeurs φ avec un coup $i \in \mathcal{D}_\varphi$ *orphelin*, c'est-à-dire que personne ne pointera plus jamais sur i dans φ :



Alors, le segment $\{\varphi(i)+1, \dots, i-1\}$ sera *invisible* pour tout le restant de la partie : personne ne l'aura dans sa vue et par visibilité personne ne pointera à l'intérieur. En conséquence, ce segment peut être extrait de φ sans changer en aucune manière les propriétés (visible, finie, bornée, ...) de φ . Ce mécanisme d'extraction induit une réécriture $\varphi \rightarrow \varphi'$ sur les structures de pointeurs.

Malheureusement, il n'est pas facile de montrer que cette réécriture termine. Dans le cas particulier où on est à profondeur bornée, on peut comme on le verra exhiber une stratégie de réduction dont on peut prouver la terminaison simplement, et en déduire le théorème de finitude. Dans le cas bien fondé c'est plus difficile, mais il reste possible d'extraire de φ une sous-structure qui est normale pour cette réduction. Or, une structure normale pour cette réécriture a également la propriété de n'avoir aucun coup orphelin : elle doit être soit finie, soit mal fondée.

Commençons par introduire quelques outils nécessaires à l'établissement de cette preuve.

Définition 4.2.11. Soit φ une structure de pointeurs, soit I un segment initial de \mathbb{N} , et $(u_i)_{i \in I} \in \mathcal{D}_\varphi^I$ une suite strictement croissante d'entiers dans le domaine de φ . Alors,

- u est une sous-structure de φ si et seulement si

$$\begin{cases} \text{pour tout } i \in I \text{ il existe } j < i \text{ tel que } u_j = \varphi(u_i) \\ \text{pour tout } i + 1 \in I, u_i \not\equiv_2 u_{i+1} \text{ (alternance)} \end{cases}$$

- Si u est une sous-structure de φ , la restriction de φ à u est définie par :

$$\varphi_u : n \mapsto \begin{cases} \perp & \text{si } n \notin I \\ m & \text{tel que } u_m = \varphi(u_n) \end{cases}$$

- u est une sous-structure visible de φ si c'est une sous-structure, et si pour tous $i, j \in I$:

$$i \in \ulcorner \varphi_u \urcorner(j) \Leftrightarrow u_i \in \ulcorner \varphi \urcorner(u_j)$$

L'ensemble des sous-structures visibles de φ est noté $VSub(\varphi)$.

Proposition 4.2.13. *On a les propriétés suivantes :*

- (1) Si φ est une structure de pointeurs ultimement visible et que $u \in VSub\varphi$, alors φ_u est une structure de pointeurs ultimement visible.
- (2) Si $(u_i)_{i \in I}$ est une sous-structure de φ et $(v_j)_{j \in J}$ est une sous-structure de φ_u , alors $(u_{v_j})_{j \in J}$ est une sous-structure de φ .
- (3) Si $(u_i)_{i \in I} \in VSub(\varphi)$ et $(v_j)_{j \in J} \in VSub(\varphi_u)$, alors $(u_{v_j})_{j \in J} \in VSub(\varphi)$.

Démonstration. (1). Soit i l'indice de visibilité de φ , et j le plus petit entier tel que $u_j \geq i$. Si un tel entier n'existe pas, c'est que nécessairement φ_u est finie et donc ultimement visible. Autrement, φ_u est visible à partir de j . En effet, pour tout $k \geq j$,

$$\begin{aligned} \varphi_u(k) \in \ulcorner \varphi_u \urcorner(k) &\Leftrightarrow u_{\varphi_u(k)} \in \ulcorner \varphi \urcorner(k) \\ &\Leftrightarrow \varphi(k) \in \ulcorner \varphi \urcorner(k) \end{aligned}$$

(2). Soit $j \in J$, il faut montrer que $\phi(u_{v_j})$ est dans $(u_{v_j})_{j \in J}$. Pour commencer, il existe $j' \in J$ tel que $v_{j'} = \phi_u(v_j)$, puisque v est une sous-structure de φ_u . Par définition de φ_u , on en déduit que $u_{v_{j'}} = \phi(u_{v_j})$, donc $\phi(u_{v_j})$ est bien dans $(u_{v_j})_{j \in J}$.

(3). Montrons que $(u_{v_j})_{j \in J}$ est une sous-structure visible de φ . On raisonne par équivalence :

$$\begin{aligned} i \in \ulcorner \varphi_u \urcorner(j) &\Leftrightarrow v_i \in \ulcorner \varphi_u \urcorner(v_j) \\ &\Leftrightarrow u_{v_i} \in \ulcorner \varphi \urcorner(u_{v_j}) \end{aligned}$$

□

Tous les raisonnements de cette section sur les structures de pointeurs reposent lourdement sur le lemme suivant :

Lemme 4.2.14 (Lemme fondamental). *Supposons que φ est une structure de pointeurs ultimement visible, et soit i un indice de visibilité pour φ . Alors, pour tout $j > i$,*

$$\exists k \in \{i + 1, \dots, j - 1\}, \varphi(k) = i \implies \ulcorner \varphi \urcorner(j) \cap \{x \mid x \not\equiv_2 j\} \cap \{\varphi(i), \dots, i - 1\} = \emptyset$$

Démonstration. Par induction sur j . Pour commençons, supposons que $j = i + 1$. Alors :

$$\begin{aligned}\lceil \varphi \rceil(i+1) &= \{i+1\} \cup \lceil \varphi \rceil(i) \\ &= \{i+1, i, \varphi(i)\} \cup \lceil \varphi \rceil(\varphi(i)-1)\end{aligned}$$

On en déduit :

$$\lceil \varphi \rceil(i+1) \cap \{x \mid x \not\equiv_2 i+1\} \cap \{\varphi(i), \dots, i-1\} = \emptyset$$

Supposons maintenant que $j > i + 1$, et que $\nexists k \in \{i+1, \dots, j-1\}$, $\varphi(k) = i$. Alors :

$$\begin{aligned}\lceil \varphi \rceil(j) \cap \{\varphi(i), \dots, i-1\} &= (\{j\} \cup \lceil \varphi \rceil(j-1)) \cap \{\varphi(i), \dots, i-1\} \\ &= \lceil \varphi \rceil(j-1) \cap \{\varphi(i), \dots, i-1\} \\ &= \lceil \varphi \rceil(\varphi(j-1)) \cap \{\varphi(i), \dots, i-1\}\end{aligned}$$

Puisque φ est supposée visible après i , on sait que $\varphi(j-1) \in \lceil \varphi \rceil(j-1)$. Par conséquent, l'hypothèse d'induction assure que $\varphi(j-1) \notin \{\varphi(i), \dots, i-1\}$. De plus, par hypothèse $\varphi(j-1) \neq i$, donc $\varphi(j-1) \notin \{\varphi(i), \dots, i\}$.

Deux cas sont désormais possibles :

- Soit $\varphi(j-1) < \varphi(i)$ et

$$\lceil \varphi \rceil(\varphi(j-1)) \cap \{\varphi(i), \dots, i-1\} = \emptyset$$

- Soit $\varphi(j-1) \in \{i+1, \dots, j-2\}$. Dans ce cas on sait par hypothèse d'induction que :

$$\lceil \varphi \rceil(\varphi(j-1)) \cap \{x \mid x \not\equiv_2 \varphi(j-1)\} \cap \{\varphi(i), \dots, i-1\} = \emptyset$$

Et puisque $\{x \mid x \not\equiv_2 \varphi(j-1)\} = \{x \mid x \not\equiv_2 j\}$, on en déduit que

$$\lceil \varphi \rceil(j) \cap \{x \mid x \not\equiv_2 j\} \cap \{\varphi(i), \dots, i-1\} = \emptyset \quad \square$$

Définition 4.2.12. Soit φ une structure de pointeurs. $i \in \mathcal{D}_\varphi$ est orphelin si et seulement si pour tout $j > i$, $\varphi(j) \neq i$.

L'existence d'un coup orphelin permet une simplification du Lemme 4.2.14, qui sera utilisée à de nombreuses reprises.

Lemme 4.2.15. Soit φ une structure de pointeurs ultimement visible et i son indice de visibilité. Supposons j orphelin dans φ , tel que $j \geq i$. Alors :

1. Pour tout $k > j$, $\lceil \varphi \rceil(k) \cap \{x \mid x \not\equiv_2 j\} \cap \{\varphi(j), \dots, j\} = \emptyset$
2. Quel que soit $k > j$, $\varphi(k) \notin \{\varphi(j), \dots, j\}$

Démonstration. (1). Application immédiate du Lemme 4.2.14.

- (2). Par (1) et visibilité de φ . □

Définition 4.2.13. Soit φ une structure de pointeurs, j un coup orphelin dans φ . Alors, on définit la suite $(\varphi \setminus j)$ de φ comme suit :

$$\begin{cases} (\varphi \setminus j)_n = n & \text{si } n < \varphi(j) \\ (\varphi \setminus j)_n = n + j - \varphi(j) + 1 & \text{sinon} \end{cases}$$

On définit alors la relation \rightarrow entre les suites d'éléments de \mathcal{D}_φ par :

$$(u_i)_{i \in I} \rightarrow (v_j)_{j \in J} \Leftrightarrow \exists k \in \mathcal{D}_{\varphi_u}, v = (\varphi_u \setminus k)$$

Lemme 4.2.16. *Supposons que φ est une structure de pointeurs ultimement visible avec i son indice de visibilité, soit $u \in VSub(\varphi)$, et j orphelin dans φ . Alors $(\varphi \setminus j) \in VSub(\varphi)$.*

Démonstration. Commençons par démontrer que $(\varphi \setminus j)$ est une sous-structure de φ . Pour commencer, elle est alternée car on a enlevé un segment de longueur paire.

Soit maintenant k un indice de $(\varphi \setminus j)$. On doit montrer que $\varphi((\varphi \setminus j)_k)$ est dans $(\varphi \setminus j)$. C'est une application immédiate du Lemme 4.2.15, puisqu'on sait que $\varphi((\varphi \setminus j)_k) \notin \{\varphi(j), \dots, j\}$.

Montrons maintenant que c'est une sous-structure visible. Par souci de lisibilité, fixons la notation $u = (\varphi \setminus j)$. On va en fait montrer la proposition légèrement plus forte suivante. Pour tout $u_k > j$:

$$\lceil_{\varphi} (u_k) \setminus \{\varphi(j), j\} = \{u_x \mid x \in \lceil_{\varphi_u} (k)\}$$

Ainsi que l'énoncé correspondant, en remplaçant les vues par des co-vues. On procède par induction sur k . Si $k < \varphi(j)$, c'est évident puisque φ et φ_u coïncident. Si $k = \varphi(j)$, alors $u_k = j + 1$. Dans ce cas, on a :

$$\begin{aligned} \lceil_{\varphi} (u_k) &= \{j + 1, j, \varphi(j), \varphi(j) - 1\} \cup \lceil_{\varphi} (\varphi(j) - 1) \\ \lceil_{\varphi_u} (k) &= \{\varphi(j) - 1\} \cup \lceil_{\varphi_u} (\varphi(j) - 1) \end{aligned}$$

Et les deux membres droits sont bien égaux à $\{\varphi(j), j\}$ près et par hypothèse d'induction. Pour la co-vue, c'est évident puisqu'on aura forcément $\varphi(u_k) < \varphi(j)$, et les vues/co-vues coïncident avant $\varphi(j)$. Si $u_k > j + 1$, alors soit une étape de calcul de vue/co-vue reste au dessus de $j + 1$ et on conclut par hypothèse d'induction, soit on passe au dessus du segment comme dans le cas précédent, et l'hypothèse d'induction conclut également. \square

À ce point, on peut donner une preuve directe du théorème de finitude, qui utilise le fait qu'une certaine stratégie de réduction pour \rightarrow est normalisante.

Théorème 3 (Finitude faible). *Soit φ une structure de pointeurs ultimement visible bornée, alors φ est finie.*

Démonstration. Si φ est ultimement visible et bornée, c'est par la Proposition 4.1.5 qu'elle est à profondeur et à mémoire finies. Soit d sa profondeur, on va montrer le théorème par induction sur d . Si $d = 0, 1$, φ est nécessairement finie. Sinon, tous les coups de φ à profondeur maximale $d + 1$ sont orphelins. Considérons la suite x_1, x_2, \dots de ces coups orphelins. On commence par remarquer que les segments $[\varphi(x_i), \dots, x_i]$ ne s'intersectent pas, par le Lemme 4.2.15. De même, si φ a une taille de vue maximale de M , il ne peut pas y avoir plus de M tels segments consécutifs. En effet, si on a :

$$[\varphi(x_i), \dots, x_i][\varphi(x_{i+1}), \dots, x_{i+1}] \dots [\varphi(x_{i+M}), \dots, x_{i+M}]$$

Alors $\forall 0 \leq k \leq M$, $x_{i+k} \in \lceil_{\varphi} (x_{i+M})$, ce qui est absurde, puisque ça implique que $\lceil_{\varphi} (x_{i+M})$ a au moins $M + 1$ éléments.

On applique alors \rightarrow à tous ces coups orphelins de profondeur maximale, dans l'ordre croissant :

$$(i)_{i \in \mathcal{D}_{\varphi}} \rightarrow u_1 \rightarrow u_2 \rightarrow \dots$$

Cette chaîne de réduction laisse invariants des préfixes de plus en plus grands, parce qu'elle ne peut pas rester constante plus de M étapes par l'argument ci-dessus. Par conséquent, elle converge une structure de pointeurs ultimement visible u_{ω} . Mais cette partie a des vues de

taille bornée, et est de profondeur d donc par hypothèse d'induction, elle est finie. On en conclut que nécessairement, φ était finie, sinon elle terminerait par une infinité de segments consécutifs, ce qui est impossible. \square

Le cœur de cette preuve est le fait qu'on a une stratégie de réduction pour \rightarrow qui termine : il s'agit d'éliminer en premier les coups orphelins de profondeur maximale. En fait, on peut utiliser les outils introduits précédemment pour démontrer la version plus forte du théorème. L'idée est de ne plus éliminer les coups orphelins par \rightarrow , mais d'utiliser la bonne fondaison de \prec pour construire directement une sous-structure de φ qui est une forme normale pour \rightarrow . C'est ce que fait le lemme suivant :

Lemme 4.2.17. *Soit φ une structure de pointeurs infinie ultimement visible, sur laquelle \prec est bien fondé. Alors il existe $u \in VSub\varphi$ infinie normale pour \rightarrow (sans coup orphelin).*

Démonstration. On définit u par induction, en maintenant l'invariant qu'il existe une extension infinie du préfixe actuellement défini qui est une sous-structure visible de φ . Pour commencer, on pose $u_i = i$ pour tout i plus petit ou égal à l'indice de visibilité de φ . Ce préfixe vérifie l'invariant, puisque $(i)_{i \in \mathcal{D}_\varphi} \in VSub(\varphi)$. Ensuite, supposons u construit jusqu'à n . Par hypothèse d'induction, on sait qu'il existe un ensemble non vide $\{v_1, v_2, \dots\}$ d'extensions infinies de u dans $VSub(\varphi)$. Dans l'ensemble $\{(v_1)_{n+1}, (v_2)_{n+1}, \dots\}$ il existe un plus petit élément pour \prec , puisque \prec est bien fondé. Supposons que ce soit $(v_i)_{n+1}$, on pose alors $u_{n+1} = (v_i)_{n+1}$ qui vérifie l'invariant d'induction par construction.

Alors, la sous-structure $u \in VSub(\varphi)$ est telle que φ_u est sans coup orphelin plus loin que l'indice de visibilité i . En effet, si $j > i$ était orphelin dans φ_u , on aurait par réduction une sous-structure $v = (\varphi_u \setminus j)$ donnant lieu à une sous-structure infinie visible $(u_{v_i})_{i \in \mathbb{N}}$ de φ avec

$$u_{v_{\varphi(j)}} \prec u_{\varphi(j)}$$

Ce qui est impossible par construction de u : le choix de $u_{\varphi(j)}$ n'aurait pas été, comme requis, minimal. \square

Théorème 4 (Finitude forte). *Soit φ une structure de pointeurs ultimement visible telle que \prec_φ est bien fondée. Alors φ est finie.*

Démonstration. Supposons par l'absurde que φ est infinie. Par le Lemme 4.2.17, on sait que φ a une sous-structure infinie sans coup orphelin. Mais dans ce cas, φ a nécessairement une chaîne de pointeurs infinie, ce qui est en contradiction avec le fait que \prec_φ est bien fondée. C'est donc que φ était finie. \square

Mentionnons une formulation équivalente du théorème, qui se révèle utile dans certains cas :

Corollaire 4.2.18. *Soit φ une structure de pointeurs infinie ultimement visible, et à profondeur bien fondée. Alors φ contient une fourche consciente infinie.*

Démonstration. Application immédiate du théorème de finitude forte et de la Proposition 4.1.5. \square

Réécriture et β -réduction. À ce point, il est naturel d'examiner les connexions possibles entre cette méthode de preuve du théorème de finitude, et celle donnée précédemment par réalisabilité sur les agents. Par exemple, on peut se demander quel effet la réécriture \rightarrow sur les structures de pointeurs induit sur les agents. Il semble bien qu'il y en ait une, mais la structure des agents actuelle est trop imprécise pour la voir. Imaginons donc qu'on donne une notion d'agent non réduite à une borne sur la taille des vues⁶, et qu'on impose que \rightarrow supprime d'un coup tous les coups orphelins qui apparaissent avec la même P -vue. Alors, si φ est une trace d'exécution d'un tel agent a , il semble que \rightarrow produise une structure de pointeur φ' trace d'exécution de a' tel que $a \rightarrow_{\beta} a'$ ⁷. Une étude de ce genre serait passionnante : d'abord parce qu'elle permettrait de relier formellement les deux méthodes de preuve du théorème de finitude données ci-dessus, mais aussi parce qu'il semble qu'on ait à portée un théorème reliant le masquage habituel en sémantique de jeux (qui peut être décomposé en étapes élémentaires de \rightarrow , au moins dans des cas particuliers) à la β -réduction en λ -calcul, donnant par là une vision nouvelle du mécanisme de l'interaction innocente en sémantique de jeux. Ces directions de recherche n'ont pas encore été suivies faute de temps, mais le seront certainement dans le futur.

4.3 Conséquences

Ici, on va passer en revue différentes conséquences du théorème de finitude. Pour commencer, on va examiner quelques généralisations ou variantes dans les structures de pointeurs, pour leur intérêt intrinsèque ou pour leur utilité dans la démonstration des théorèmes de composition.

4.3.1 Résultats intermédiaires

Finitude quantitative. Cette généralisation du théorème de finitude sera essentielle dans la preuve de composition des stratégies bornées.

Lemme 4.3.1. *Si $n, p \in \mathbb{N}$, alors il existe une borne $T(n, p) \in \mathbb{N}$ telle que, pour tout $\varphi \in n \star p$,*

$$|\varphi| \leq T(n, p)$$

Démonstration. L'ensemble $n \star p$ peut être vu comme un arbre en considérant l'ordre préfixe, et cet arbre est clairement à branchement fini : en effet, à chaque étape on doit pointer dans l'histoire qui est toujours finie. Une branche infinie pour cet arbre serait une structure de pointeurs infinie $\varphi \in n \star p$, ce qui est interdit par le théorème de finitude faible. Par conséquent, par le Lemme de König, l'arbre considéré est fini : on pose $T(n, p)$ sa profondeur ; c'est la longueur maximale d'une structure de $n \star p$. \square

Il serait intéressant de trouver une valeur plus concrète pour $T(n, p)$, mais ça sort malheureusement du cadre de cette thèse.

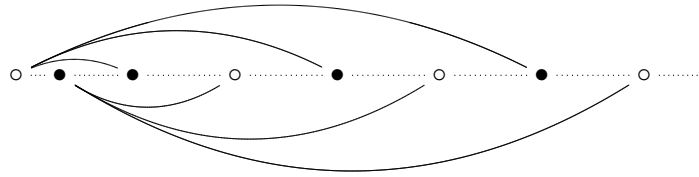
⁶De tels agents ressembleraient à des fermetures M^{σ} où M est β -normal et η -long et où toutes les fermetures dans σ vérifient encore ces conditions.

⁷Il faut bien sûr étendre β aux agents ce qui est loin d'être immédiat, en particulier il faut une restriction de β bien choisie qui préserve la structure des agents esquissée dans la note précédente.

Unicité du coupable. Comme évoqué à la fin de la Section 4.1.2, le coupable est unique dans les parties à profondeur bien fondée : seul un des deux joueurs peut avoir une fourche consciente infinie.

Lemme 4.3.2. *Soit φ une structure de pointeurs à profondeur bien fondée (i.e. elle n'admet pas de chaîne de pointeurs infinie). Alors si on a deux fourches conscientes infinies $(\psi_0, (\psi_i)_{i \in \mathbb{N}_*})$ et $(\psi'_0, (\psi'_i)_{i \in \mathbb{N}_*})$, on a $\psi_0 \equiv_2 \psi'_0$.*

Démonstration. Supposons par l'absurde qu'on ait deux fourches conscientes comme dans l'énoncé, avec $\psi_0 \not\equiv_2 \psi'_0$. On va d'abord en extraire deux fourches *intriquées*, comme illustré dans la figure ci-dessous :



On le fait en construisant par induction deux suites d'entiers σ et τ comme suit. Supposons sans perte de généralité que $\psi_1 < \psi'_1$, l'autre cas est symétrique. On pose $\sigma(1) = 1$. Supposons qu'on vienne de définir $\sigma(i)$ alors on définit $\tau(i)$ comme le plus petit j tel que $\psi'_j > \psi_{\sigma(i)}$. Si on vient de définir $\tau(i)$, on définit $\sigma(i+1)$ comme le plus petit j tel que $\psi_j > \psi_{\tau(i)}$. On obtient deux fourches conscientes infinies $(\psi_0, (\psi_{\sigma(i)})_{i \in \mathbb{N}_*})$ et $(\psi'_0, (\psi'_{\tau(i)})_{i \in \mathbb{N}_*})$ qui sont bien intriquées, c'est-à-dire que pour tout i , on a

$$\psi_{\sigma(i)} < \psi'_{\tau(i)} < \psi_{\sigma(i+1)} < \psi'_{\tau(i+1)}$$

Considérons maintenant un coup orphelin j dans un segment $[\psi_{\sigma(i)} \dots \psi_{\tau(i)}]$ et ne participant pas aux fourches, c'est-à-dire que j et $\varphi(j)$ ne sont ni des ψ_i ni des ψ'_i . Alors le segment $[\varphi(j) \dots j]$ est strictement inclus dans le segment $[\psi_{\sigma(i)} \dots \psi_{\tau(i)}]$. En effet, si $\varphi(j) < \psi_{\sigma(i)}$, $\psi_{\sigma(i)}$ ne sera plus jamais visible par le Lemme 4.2.15, ce qui est impossible puisqu'il fait partie d'une fourche consciente, en particulier $\psi_{\sigma(i)} \in \ulcorner \varphi \lrcorner (\psi_{\sigma(i+1)})$, par conséquent

$$[\varphi(j) \dots j] \subset [\psi_{\sigma(i)} \dots \psi_{\tau(i)}]$$

L'inclusion est stricte pour les deux bornes puisqu'on a supposé que j et $\varphi(j)$ ne participaient pas aux fourches. Considérons maintenant $u_1 = (\varphi \setminus j)$. C'est une sous-structure visible de φ qui possède la caractéristique de conserver les mêmes fourches conscientes intriquées (à réindexation près).

On itère l'opération ci-dessus en commençant par les plus petits tels coups orphelins internes aux fourches intriquées, pour obtenir une suite de sous-structures de pointeurs u_1, u_2, \dots de φ contenant les deux fourches. Mais cette suite converge : en effet, chaque intervalle $[\psi_{\sigma(i)} \dots \psi_{\tau(i)}]$ est fini, donc à partir d'un certain rang on otera plus de coup à l'intérieur. Par conséquent, la suite u_1, \dots laisse invariants des préfixes de plus en plus longs de φ , et converge vers une sous-structure u_ω de φ . Or, les fourches intriquées existent encore dans φ_{u_ω} , mais il n'y a plus de coups orphelins à l'intérieur. On part d'un coup interne aux fourches, et on suit les pointeurs en sens inverse, ce qui ne peut mener qu'à d'autres coups internes aux fourches : on construit ainsi une chaîne de pointeurs infinie, ce qui est en contradiction avec l'hypothèse. \square

Parties non alternées. À plusieurs reprises, des extensions de la notion de partie légale au cas non alterné ont été considérées, citons par exemple le modèle du π -calcul asynchrone [Lai05] de Laird, utilisant comme structure de base des parties non alternées. Bien que peu explicité, il existe un exemple plus universel : parfois, il peut être utile de voir une interaction dans $I(A, B, C)$ comme une partie (pas nécessairement alternée) sur l'arène $(A \Rightarrow B) \Rightarrow C$. Une telle opération peut paraître sémantiquement (et esthétiquement !) discutable, mais est cependant confortée par la *sémantique des jeux découverte* [Blu08] : il est parfaitement viable de considérer l'interaction sans le «masquage» qui lui est en général associé, ce qui amène à considérer les interactions dans $I(A, B, C)$ comme des parties légales habituelles. Indépendamment de telles considérations, ce sera utile dans notre contexte : prouver la préservation de la totalité impose de borner la longueur des interactions. Or, les interactions ne sont en général pas alternées ; il faut donc généraliser le résultat de finitude aux parties non alternées.

La complication est que les structures de pointeurs s'adaptent très mal au cas non alterné. En effet, dès lors qu'on ne suppose plus l'alternance on ne peut plus se baser sur la classe d'équivalence modulo 2 pour déterminer la polarité d'un coup. On va donc se poser le problème directement sur les parties ultimement visibles non alternées de I_ω . Si A est une arène, on obtient les parties *légales non alternées* de A en ôtant la condition d'alternance de la Définition 2.1.9. Les définitions de P -vue et de O -vue restent inchangées.

Définition 4.3.1. *Par commodité, commençons par donner quelques définitions sur les parties légales non alternées de I_ω .*

- La vue $\lceil s \rceil$ est $\lceil s \rceil$ si $\lambda_{I_\omega}(s_\omega) = P$ et $\lfloor s \rfloor$ sinon,
- On définit $i \prec_s j$ par induction, avec $i \prec_s j$ si et seulement si :
 - $\lambda_{I_\omega}(s_{i-1}) = \lambda_{I_\omega}(s_i)$, et $i - 1 = j$ ou $i - 1 \prec j$.
 - $\lambda_{I_\omega}(s_{i-1}) \neq \lambda_{I_\omega}(s_i)$, s_{i-1} pointe sur $s_{i'}$ et $i' = j$ ou $i' \prec_s j$.
- De même, une partie s sur I_ω est ultimement visible si et seulement si elle devient O -visible et P -visible à partir d'un certain rang.

Lemme 4.3.3. *Soit s une partie légale non nécessairement alternée de I_ω . Alors il existe une structure de pointeurs φ_s telle que :*

- s est ultimement visible si et seulement si φ_s est ultimement visible,
- \prec_s est bien fondée si et seulement si \prec_{φ_s} est bien fondée,
- s est finie si et seulement si φ_s est finie.

Démonstration. On va construire par induction sur n une fonction strictement croissante $\sigma_n : \{0, \dots, n\} \rightarrow \mathbb{N}$ et une structure de pointeurs φ_s de longueur $\sigma(n) + 1$ tels que :

- Pour tout $i \leq n$, $\lambda_{I_\omega}(s_i) = P$ si et seulement si $\sigma(i) \equiv_2 1$,
- Pour tous $i, j \leq n$, $i \prec_s j$ si et seulement si $\sigma(i) \prec_{\varphi_s} \sigma(j)$.

Pour amorcer la construction, on pose $\sigma(0) = 0$ et $\varphi_s(0) = 0$. Supposons qu'on ait construit φ_s et σ jusqu'à n . Si s est de longueur n , alors c'est fini. Sinon, deux cas sont possibles :

- Soit $\lambda_{I_\omega}(s_{n+1}) = \lambda_{I_\omega}(s_n)$. Dans ce cas on insère dans φ_s un coup factice pour rendre φ_s alternée. On pose $\varphi_s(\sigma(n) + 1) = \sigma(n)$. Ensuite, on pose $\sigma(n + 1) = \sigma(n) + 2$ et si s_{n+1} pointe vers s_i , on pose $\varphi_s(\sigma(n + 1)) = \sigma(i)$. Les deux invariants d'induction sont clairement préservés.
- Soit $\lambda_{I_\omega}(s_{n+1}) \neq \lambda_{I_\omega}(s_n)$. Dans ce cas on pose simplement $\sigma(n + 1) = \sigma(n) + 1$ et si s_{n+1} pointe vers s_i , on pose $\varphi_s(\sigma(n + 1)) = \sigma(i)$, ce qui préserve clairement les hypothèses d'induction.

Par construction, φ_s , vérifie bien les trois propriétés annoncées dans l'énoncé du lemme. \square

Théorème 4.3.4 (Finitude non alternée). *Si s est une partie légale éventuellement infinie, pas nécessairement alternée sur I_ω , alors si \prec_s est bien fondé, s est finie.*

Démonstration. Corollaire immédiat du Lemme 4.3.3 : si \prec_s est bien fondé, alors \prec_{φ_s} est bien fondé, donc φ_s est finie, donc s est finie. \square

Dans la suite, on utilisera la notation φ_u pour extraire la structure de pointeurs correspondant à u , que u soit ou non alternée. Dans les deux cas, la construction préserve bien toutes les propriétés basées sur la vue.

4.3.2 Stratégies finies, bornées, noéthériennes

Dans cette sous-section, on va définir trois principales classes de stratégies pour lesquelles on montrera que la totalité est préservée par composition. La première classe est celle des stratégies *finies*. Cette classe est bien connue en sémantique de jeux : c'est celle dans laquelle s'interprètent naturellement les preuves ou λ -termes typés. Ce sont les stratégies innocentes qui possèdent aussi la caractéristique d'avoir une fonction de vue finie, et ce sont également les éléments compacts associés à la structure enrichie d'ordres complets de **Inn**. La seconde classe est celle des stratégies *bornées* : elles n'ont pas nécessairement un nombre fini de P -vues, cependant la longueur de leurs vues ne peut dépasser une borne fixée. Finalement, on présentera la dernière classe des stratégies *noéthériennes* : les stratégies de cette classe peuvent avoir des vues de longueur non bornées, mais ne peuvent pas avoir de vue *infinie*. En d'autres termes, elles n'admettent ni chaîne de pointeur infinie ni fourche consciente infinie. Il est important de noter qu'entre ces trois classes de stratégies, seules les stratégies finies sont supposées innocentes.

Définition 4.3.2. *Une stratégie P -visible σ (pas nécessairement déterministe, ou innocente) est :*

- Finie si σ est innocente et si sa fonction de vue est finie,
- Bornée si il existe $N \in \mathbb{N}$ tel que pour tout $s \in \sigma$, $|\ulcorner s \urcorner| \leq N$,
- Noéthérienne Si pour tout $s \in \sigma$ (éventuellement infinie), \prec_{φ_s} est bien fondée sur les coups Joueurs (i.e. de polarité impaire) ou, de façon équivalente (Proposition 4.1.5), si φ_s n'a ni chaîne de pointeur infinie ni fourche consciente infinie appartenant à Joueur.

Remarque. Le défaut de la présentation des stratégies noéthériennes donnée au dessus est qu'elle n'est pas interne à la sémantique de jeux, c'est-à-dire qu'il faut du vocabulaire venant des structures de pointeurs pour la définir correctement. Le cœur du problème est en fait qu'en sémantique de jeux, la vue d'une partie est définie comme une sous-suite : elle n'est pas localisée. Dire, si $s_1 \sqsubseteq s_2 \sqsubseteq s$, que $\ulcorner s_1 \urcorner \sqsubseteq \ulcorner s_2 \urcorner$ n'implique pas que cette inclusion sera vérifiée dans les structures de pointeurs. Une condition suffisante et presque nécessaire pour qu'une stratégie soit noéthérienne est la suivante :

Proposition 4.3.5. *Soit σ une stratégie. Supposons qu'il n'y ait pas de chaîne de préfixes infinie $s_1 \sqsubset s_2 \sqsubset s_3 \sqsubset \dots$ dans σ telle que :*

$$\ulcorner s_1 \urcorner \sqsubset \ulcorner s_2 \urcorner \sqsubset \ulcorner s_3 \urcorner \sqsubset \dots$$

alors σ est noéthérienne.

Démonstration. Il suffit de remarquer qu'une chaîne de pointeurs infinie ou une fourche consciente infinie fournirait une telle chaîne de préfixes. \square

Ces deux définitions ne sont malheureusement pas équivalentes, comme le montre le contre-exemple ci-dessus :



Opposant crée une fourche infinie. Dans chaque copie, l'interaction va plus loin et Joueur a des vues de plus en plus grandes, on a donc bien une violation de la condition donnée dans la Proposition 4.3.5, et ce même si Joueur n'a pas de vue infinie. L'écart entre ces deux conditions reste cependant très faible. Essayons de caractériser les situations où Joueur brise la condition de la Proposition 4.3.5 sur une partie (infinie) s , sans avoir de vues infinies. Extrayons la structure de pointeurs φ_s correspondante. Pour commencer, elle n'a pas de chaîne de pointeurs infinie, donc φ_s est à profondeur bien fondée. Par le Corollaire 4.2.18, on sait donc qu'elle possède une fourche consciente infinie. Cette fourche ne peut pas appartenir à Joueur, puisque cela constituerait une vue infinie. Par conséquent, elle est à Opposant. Les seules parties acceptées par la définition directe dans les structures de pointeurs mais pas par la Proposition 4.3.5 sont donc des parties où Opposant a une fourche consciente infinie : situation qui ne peut pas arriver dans une catégorie de stratégies noéthériennes.

Compacité. Un cas particulier notable des définitions ci-dessus est le cas des arènes finies : dans ce cas, les trois notions définies ci-dessus s'écrasent, du moins pour les stratégies innocentes.

Théorème 4.3.6 (Compacité). *Soit σ une stratégie innocente sur une arène A finie. Alors, les trois propriétés suivantes sont équivalentes :*

- (1) σ est finie.
- (2) σ est bornée.
- (3) σ est noéthérienne.

Démonstration. (1) \Rightarrow (2), (2) \Rightarrow (3) trivial. Montrons (3) \Rightarrow (1), supposons σ noéthérienne, et examinons son arbre de vues. Cet arbre est à branchement fini par finitude de l'arène, et il est à profondeur finie car σ est noéthérienne. Il est donc fini par le lemme de König et σ est finie. \square

Le nom de *théorème de compacité* vient du théorème du même nom énoncé par Hyland dans [Hy197], affirmant que les stratégies respectant le *principe de justice* sont finies. C'est précisément ce résultat qu'on prouve ici, en effet le principe de justice sélectionne les stratégies qui ne possèdent pas de fourche consciente, c'est-à-dire (dans une arène finie) qu'elles sont noéthériennes. On note par contre que lorsque l'arène n'est plus finie, que ce soit en largeur ou en profondeur, ces trois classes de stratégies sont en général distinctes. Par exemple, l'identité sur l'arène des entiers plats est bornée, mais pas finie. Par ailleurs, on a déjà donné en Section 4.2 une stratégie **repeat**, qui est noéthérienne mais pas bornée.

On va maintenant montrer comment ces trois classes de stratégies (finies, bornées, noéthériennes) sont stables par composition.

Stratégies bornées. Quelques outils préliminaires sont nécessaires pour traiter le cas des stratégies bornées non nécessairement innocentes. Toutes les définitions non développées ici sont données dans la partie préliminaire.

Proposition 4.3.7. *Toute stratégie P -visible σ est bornée si et seulement si sa fermeture saturée (voir Section 2.2.3) $\bar{\sigma}$ est bornée.*

Démonstration. Il suffit de remarquer que $\lceil \sigma \rceil = \lceil \bar{\sigma} \rceil$. \square

Définition 4.3.3. *Soient $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ P -visibles. Une pré-vue de $\sigma; \tau$ est une interaction $u \in \sigma || \tau$ avec un unique coup initial dans C , tel que l'opposant externe pointe toujours vers le coup précédent, i.e. $u \upharpoonright_{A,C}$ est une P -vue.*

L'astuce ici est de considérer les pré-vues sur $I(A, B, C)$ comme des parties légales sur $(A \Rightarrow B) \Rightarrow C$, et d'étudier la structure de pointeurs induite pour déduire une borne sur la longueur de la pré-vue. La propriété fondamentale des pré-vues est la *condition de switching* : seul Joueur peut basculer entre B et C , et seul Opposant peut basculer entre A et B . En conséquence, si u se termine avec un coup dans B ou C , sa P -vue va seulement visiter B et C et sera en fait une P -vue sur $B \Rightarrow C$. De même, si u termine par un coup dans A , $\lceil u \rceil \upharpoonright_{A,B}$ sera une O -vue de $A \Rightarrow B$ où chaque coup dans A pointera vers le coup précédent. De manière duale, $\lfloor u \rfloor \upharpoonright_{A,B}$ sera une P -vue de $A \Rightarrow B$; et $\lfloor u \rfloor \upharpoonright_{B,C}$ sera une O -vue de $B \Rightarrow C$ où chaque coup dans C pointera vers le coup précédent.

Lemme 4.3.8. *Si les longueurs des P -vues de σ et τ sont bornées respectivement par N_σ et N_τ , et si u est une pré-vue de $\sigma || \tau$, alors $\varphi_u \in (N_\sigma + N_\tau) \star \max(N_\sigma + 1, N_\tau)$.*

Démonstration. Par l'analyse ci-dessus, la forme générale d'une P -vue de u est $s \cdot t$ où s est une P -vue de τ et t est une O -vue de A qui est aussi une P -vue : t doit donc être une sous-suite d'une P -vue de σ . Par conséquent, les longueurs des P -vues de φ_u sont bornées par $N_\sigma + N_\tau$.

Une O -vue de u est soit une O -vue de C qui est aussi une sous-séquence d'une P -vue de τ et donc bornée par N_τ , ou bien est de la forme $c \cdot s$, où c est un coup initial de C et s est une P -vue de σ et donc bornée par $N_\sigma + 1$. Par conséquent, les longueurs des O -vues de φ_u sont bornées par $\max(N_\sigma + 1, N_\tau)$. \square

Proposition 4.3.9. *Si $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ sont des stratégies bornées et saturées, alors $\sigma; \tau$ est bornée.*

Démonstration. Soit P l'ensemble des pré-vues de $\sigma || \tau$. Par le Lemme 4.3.8, chaque $u \in P$ génère $\varphi_u \in (N_\sigma + N_\tau) \star \max(N_\sigma + 1, N_\tau)$, et la longueur de ces structures est bornée par $T(N_\sigma + N_\tau, \max(N_\sigma + 1, N_\tau))$ par le Lemme 4.3.1.

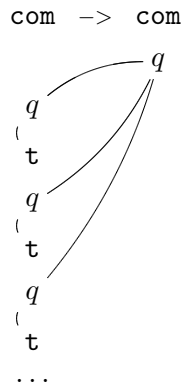
À ce point, on remarque que puisque σ et τ sont saturées, $\sigma; \tau$ est toujours saturée. Par conséquent, si $s \in \sigma; \tau$, alors $\lceil s \rceil \in \sigma; \tau$ et le raisonnement ci-dessus montre que $|\lceil s \rceil| \leq T(N_\sigma + N_\tau, \max(N_\sigma + 1, N_\tau))$, donc $\sigma; \tau$ est bornée. \square

Théorème 4.3.10. *Soient $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies bornées, alors $\sigma; \tau$ est bornée.*

Démonstration. Par la Proposition 4.3.7, $\bar{\sigma}$ et $\bar{\tau}$ sont bornées (et saturées). Par la Proposition 4.3.9, $\bar{\sigma}; \bar{\tau}$ est donc bornée. Mais par monotonie de la composition, on a :

$$\sigma; \tau \subseteq \bar{\sigma}; \bar{\tau}$$

Donc $\sigma; \tau$ est elle aussi bornée. \square

FIG. 4.7 – Début de la fourche consciente de `random; repeat`.

Stratégies finies.

Théorème 4.3.11. *Soient $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies finies, alors $\sigma; \tau : A \Rightarrow C$ est finie.*

Démonstration. Par le Théorème 4.3.10, $\sigma; \tau$ est bornée. Mais puisque σ et τ sont toutes deux finies, il existe en fait des restrictions finies A' , B' et C' de A , B et C telles que $\sigma : A' \Rightarrow B'$ et $\tau : B' \Rightarrow C'$. Par conséquent, $\sigma; \tau$ est une stratégie bornée sur l'arène finie $A \Rightarrow C$. Par le Théorème 4.3.6, elle est donc finie. \square

Stratégies noethériennes. Un fait déconcertant : les stratégies noethériennes ne sont pas stables par composition. Considérons le terme `repeat : nat -> com -> com` introduit en Section 4.2, ainsi que le terme `random : nat` qui peut renvoyer n'importe quel entier, mais *doit* en renvoyer un. Les deux stratégies correspondantes sont noethériennes. Cependant, `random; repeat` possède une fourche consciente infinie, comme illustré dans la Figure 4.7. La raison en est simple : chaque préfixe de cette partie est dans une interaction possible entre `random` et `repeat` ; il suffit que `random` choisisse un nombre assez grand. Manifestement, ce phénomène n'arriverait pas si `random` devait faire son choix dans un ensemble borné. C'est ce qu'on va montrer avec un résultat de préservation par composition des stratégies noethériennes et *finiment branchantes*.

Définition 4.3.4. *Une stratégie σ est finiment branchante si pour tout $s \in \sigma$, pour toute extension sa de s , l'ensemble $\{sab \mid sab \in \sigma\}$ est fini.*

Lemme 4.3.12. *Soit $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies noethériennes, soit $u \in \sigma \parallel \tau$, potentiellement infini. Alors si Opposant n'a pas de fourche consciente infinie dans $u \upharpoonright_{A,C}$, u est finie.*

Démonstration. Commençons par noter que u est à profondeur bien fondée : c'est clair parce que toute chaîne de pointeur infinie finirait dans A , B ou C et serait soit dans σ , soit dans τ . C'est impossible puisqu'elles sont supposées noethériennes.

On va montrer que u est finie. Pour ça, on va montrer qu'en fait, u ne contient pas de fourche consciente. Supposons par l'absurde que φ_u contienne une fourche consciente ψ . Notons que u n'était pas nécessairement alternée, donc φ_u est obtenue comme décrit plus haut dans

cette sous-section. Cependant, ψ_0 était forcément déjà dans u , puisque la construction de φ_u ne fait pointer personne vers les coups ajoutés. Puisqu'on a supposé qu'Opposant n'avait pas de fourche consciente infinie dans $u \upharpoonright_{A,C}$, ψ_0 correspond soit à un coup Opposant dans $u \upharpoonright_{B,C}$ (impossible car τ est noéthérienne), soit à un coup Opposant dans $u \upharpoonright_{A,B}$ (impossible car σ est noéthérienne). Donc φ_u n'a pas de fourche consciente et étant à profondeur finie, elle est finie par le Corollaire 4.2.18, donc u est finie. \square

Théorème 4.3.13. *Soient $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies noéthériennes et finiment branchantes, alors $\sigma; \tau$ est encore finiment branchante et noéthérienne.*

Démonstration. Montrons que $\sigma; \tau$ est finiment branchante. Soit $s \in \sigma; \tau$, et sa une extension immédiate. Considérons l'ensemble $\{u \in \sigma \parallel \tau \mid \exists b, u \upharpoonright_{A,C} = sab\}$: on lui donne une structure d'arbre par l'ordre préfixe. Cet arbre est à branchement fini, puisque σ et τ sont à branchement fini. De plus, cet arbre n'a pas de branche infinie, par le Lemme 4.3.12. En effet, Opposant ne peut avoir de fourche consciente infinie sur $u \upharpoonright_{A,C}$ avec $u \upharpoonright_{A,C} = sab$ fini. Par conséquent cet arbre est fini, et l'ensemble des extensions de $\sigma; \tau$ à sa est fini.

Montrons maintenant que $\sigma; \tau$ est noéthérienne. Pour commencer, $s \in \sigma; \tau$ potentiellement infinie ne peut pas avoir de chaîne de pointeurs infinie, car celle-ci serait nécessairement dans σ ou τ . Si Opposant a une fourche consciente infinie dans s , alors Joueur ne peut pas en avoir par le Lemme 4.3.2. D'un autre côté, si Opposant n'a pas de telle fourche, Joueur ne peut pas en avoir : cela donnerait un témoin infini $u \in \sigma \parallel \tau$ qui serait en contradiction avec le Lemme 4.3.12. \square

Ce résultat de composition des stratégies noéthériennes et finiment branchantes est notable, puisqu'aucune de ces deux conditions n'est stable à elle seule par composition. On a vu plus haut que les stratégies noéthériennes ne composent pas, c'est aussi vrai pour les stratégies qui sont seulement finiment branchantes. Soit `rand_bool()` une stratégie renvoyant aléatoirement `true` ou `false`. considérons l'interprétation du terme suivant, représentée en Figure 4.8.

$$T = \lambda F n. \text{ if rand_bool() then } n \text{ else } F (n + 1)$$

On peut constater que $\llbracket T \rrbracket$ est bien finiment branchante. On la précompose avec l'opérateur de point fixe $\llbracket Y \rrbracket$, pour obtenir la stratégie représentée en Figure 4.9, qui est clairement non finiment branchante.

Préservation de la totalité. Montrons maintenant que pour toutes ces stratégies, la totalité est préservée par composition.

Théorème 4.3.14. *Soit $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies finies (resp. bornées, noéthériennes) et totales, alors $\sigma; \tau$ est totale.*

Démonstration. Supposons qu'il y ait $s \in \sigma; \tau$, et une extension sa tel qu'il n'y a pas de $sab \in \sigma; \tau$. Par la propriété du témoin, il existe $u \in \sigma \parallel \tau$ tel que $u \upharpoonright_{A,C} = s$. Supposons (sans perte de généralité) que a est dans A . Alors, puisque σ est totale, il existe b tel que $uab \in \sigma \parallel \tau$. Mais b est nécessairement dans B , sinon on aurait une extension à sa dans $\sigma; \tau$. Le même raisonnement tient ensuite pour τ , puis σ , etc. . . On construit ainsi une partie infinie $u \in \sigma \parallel \tau$. Mais $u \upharpoonright_{A,C} = sa$ est fini, en particulier Opposant ne peut y avoir de fourche consciente infinie, donc u est finie par le Lemme 4.3.12 : contradiction. \square

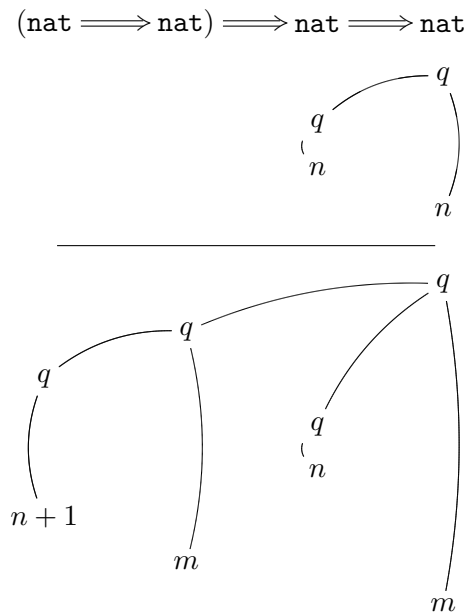


FIG. 4.8 – *P*-vues de $\llbracket T \rrbracket$

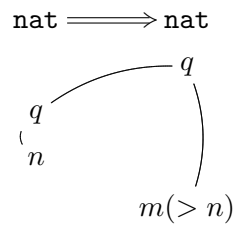


FIG. 4.9 – *P*-vues de $\llbracket Y \rrbracket; \llbracket T \rrbracket$

Deuxième partie

Points fixes en Sémantique des Jeux

Chapitre 5

Une logique avec points fixes : μLJ

Sommaire

5.1 Règles	132
5.1.1 Formules, séquents	132
5.1.2 Fragments	135
5.2 Réduction	136
5.2.1 Règles de réduction	136
5.2.2 Coupures irréductibles	146
5.3 Expressivité	148
5.3.1 Quotient structurel	148
5.3.2 Rappel : définition du système T	149
5.3.3 Traduction dans μLJ	150

Les définitions inductives et coinductives dans la syntaxe ont été étudiées dans un grand nombre de systèmes, comme la logique linéaire [BM07], le λ -calcul [AA99], le système F [Mat98] ou la théorie des types de Martin-Löf [CD94]. La motivation première est bien sûr de modéliser le raisonnement par induction naturel en mathématiques. Par ailleurs, ces constructions permettent souvent d'augmenter drastiquement l'expressivité d'une logique sans payer le prix des modalités exponentielles (voir [BM07]) ou de l'imprédictivité (voir [AA99]). Le langage étudié dans cette partie s'inscrit dans cette lignée. Il s'agit de la forme propositionnelle de μLJ [BM07], un calcul des séquents intuitionniste avec des constructeurs pour le raisonnement inductif et coinductif. Cette logique peut être vue de nombreuses façons, chacune avec son lot d'intuitions.

Extension de LJ . Bien sur, μLJ peut être vue comme une extension naturelle de LJ . Dans ce contexte, les formules de μLJ sont des propositions logiques sur des objets éventuellement infinis, sur lesquels on peut raisonner par induction ou coinduction. Cette vision de μLJ le place comme une première étape dans la construction de langages plus puissants comme COQ, capables de formaliser une partie des mathématiques.

Extension du μ -calcul. On peut également la voir comme la logique obtenue par l'ajout du connecteur \Rightarrow au μ -calcul [AN01] propositionnel. C'est cette vision qui inspirera la sémantique des jeux à venir. En effet, il existe une représentation très précise [San02b] des μ -termes par des jeux, équipés de conditions de gains appelés de *parité*. Il semble donc naturel d'étendre les

conditions de gain des jeux de parité aux jeux d'arènes et stratégies innocentes ; c'est d'ailleurs l'objectif de cette partie.

Un langage de programmation. Finalement, on peut aussi voir μLJ comme un langage abstrait, très explicite (en particulier, le style multiplicatif adopté autorise une gestion fine des règles structurelles), dans lieux ni syntaxe concrète. Ce langage n'est bien sûr pas Turing-puissant, mais a cependant une grande expressivité : on verra notamment qu'on peut y encoder le système T de Gödel [GTL89]. Indépendamment des fonctions représentées, la syntaxe des types de μLJ permet une représentation très souple et puissante des structures de données non bornées (entiers, listes, arbres) ou paresseuses (flots).

5.1 Règles

5.1.1 Formules, séquents.

On définit d'abord les pré-formules et séquents de μLJ .

Définition 5.1.1. *Soit \mathcal{V} un ensemble de variables. Les pré-formules de μLJ sont données par la grammaire suivante, avec $X \in \mathcal{V}$.*

$$S ::= S \Rightarrow T \mid S + T \mid S \times T \mid \mu X.T \mid \nu X.T \mid X \mid 1 \mid 0$$

Pour le moment, cette classe de formules est trop générale. Tout d'abord, la normalisation et la cohérence de la logique ne peuvent être espérées que si l'on se restreint aux formules *positives*, *i.e.* on ne peut appliquer μ ou ν que sur des variables libres qui n'apparaissent qu'en position covariante : en effet, il serait facile dans le cas contraire de typer l'analogue dans μLJ de $(\lambda x.xx)(\lambda x.xx)$. On définira donc les *formules* de μLJ comme les pré-formules satisfaisant cette contrainte. Par ailleurs, l'interprétation en sémantique de jeux imposera pour des raisons techniques de se restreindre à un fragment de μLJ qui sera introduit un peu plus tard.

Définition 5.1.2. *Soit F une pré-formule de μLJ . On définit par induction mutuelle ses variables positives et ses variables négatives comme suit. La définition des variables positives :*

$$\begin{aligned} fv^+(S \Rightarrow T) &= fv^-(S) \cup fv^+(T) \\ fv^+(S + T) &= fv^+(S) \cup fv^+(T) \\ fv^+(S \times T) &= fv^+(S) \cup fv^+(T) \\ fv^+(\mu X.T) &= fv^+(T) \setminus \{X\} \\ fv^+(\nu X.T) &= fv^+(T) \setminus \{X\} \\ fv^+(X) &= \{X\} \\ fv^+(1) &= \emptyset \\ fv^+(0) &= \emptyset \end{aligned}$$

Et celle des variables négatives :

$$\begin{aligned}
fv^-(S \Rightarrow T) &= fv^+(S) \cup fv^-(T) \\
fv^-(S + T) &= fv^-(S) \cup fv^-(T) \\
fv^-(S \times T) &= fv^-(S) \cup fv^-(T) \\
fv^-(\mu X.T) &= fv^-(T) \\
fv^-(\nu X.T) &= fv^-(T) \\
fv^-(X) &= \emptyset \\
fv^-(1) &= \emptyset \\
fv^-(0) &= \emptyset
\end{aligned}$$

Définition 5.1.3. Les formules de μLJ sont définies par induction parmi les pré-formules, comme suit :

- $F = X$ est une formule,
- 0 et 1 sont des formules,
- Si S et T sont des formules, $F = S \Rightarrow T$ (resp. $S \times T$ et $S + T$) est une formule si $fv^+(F) \cap fv^-(F) = \emptyset$.
- Si S est une formule et $X \notin fv^-(S)$, alors $\mu X.S$ et $\nu X.S$ sont des formules.

Une formule F est close si $fv^+(F) \cup fv^-(F) = \emptyset$.

Définition 5.1.4. Un foncteur positif est une formule F de μLJ telle que $fv^-(F)$ est vide et $fv^+(F)$ est un singleton. Les foncteurs positifs seront désignés par la lettre P , et quand ça ne causera pas d'ambiguïté on désignera par X leur variable libre. On notera alors $P(A) = P[A/X]$.

Les foncteurs négatifs sont donnés par la définition duale. On les désignera par N , et on prendra les mêmes conventions.

Définition 5.1.5. Un séquent est une expression de la forme $\Gamma \vdash A$, où Γ est une liste de formules et A est une formule.

Règles de déduction. On donne en Figure 5.1.1 les règles de déduction de μLJ , séparées comme souvent en plusieurs groupes. Les groupes d'identité, structurel et logique sont les groupes habituels dans le calcul des séquents. Les nouveautés ici sont les groupes de *points fixes*, donnant les règles de déduction sur les plus petits et plus grands points fixes. On remarquera également l'ajout de règles de *foncteurs*, autorisant à considérer les formules non closes de μLJ comme des foncteurs et d'effectuer des déductions en conséquence. Ces règles ne sont pas essentielles : en particulier elles ne changent rien à la prouvabilité (on verra qu'on peut les éliminer), cependant, elles facilitent considérablement l'expression des règles de réduction. Notons enfin qu'à part pour ce qui est des règles foncteur, toutes les formules considérées sont supposées être *closes*.

Règles affaiblies. Au cours de cette thèse, on pourra parler des *règles affaiblies* de μLJ . Il s'agit des règles de la Figure 5.1.1, mais sans le contexte Γ pour les règles des groupes Points Fixes et Foncteurs. C'est la logique μLJ affaiblie qui se rapproche le plus du modèle catégorique des points fixes donné par les algèbres initiales et coalgèbres terminales. Notons que c'est pour cette logique affaiblie que l'auteur a initialement construit un modèle [Cla09a], qui a ensuite été généralisé aux règles générales dans [Cla09b].

Groupe Identité
$\frac{}{A \vdash A} \text{ax} \qquad \frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta \vdash B} \text{Cut}$
Groupe Structurel
$\frac{\Gamma, A, A, \Delta \vdash B}{\Gamma, A, \Delta \vdash B} C \qquad \frac{\Gamma, \Delta \vdash B}{\Gamma, A, \Delta \vdash B} W \qquad \frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C} \gamma$
Groupe Logique
$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B} \Rightarrow_r \qquad \frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Gamma, \Delta, A \Rightarrow B \vdash C} \Rightarrow_l$ $\frac{}{0 \vdash A} 0_l \qquad \frac{\Gamma \vdash A}{\Gamma, 1 \vdash A} 1_l \qquad \frac{}{\vdash 1} 1_r$ $\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \times B} \times_r \qquad \frac{\Gamma, A, B \vdash C}{\Gamma, A \times B \vdash C} \times_l$ $\frac{\Gamma \vdash A}{\Gamma \vdash A + B} \leftarrow_{\vdash_r} \qquad \frac{\Gamma \vdash B}{\Gamma \vdash A + B} \rightarrow_{\vdash_r} \qquad \frac{\Gamma, A \vdash C \quad \Delta, B \vdash C}{\Gamma, \Delta, A + B \vdash C} +_l$
Points Fixes
$\frac{\Gamma \vdash P(\mu X.P)}{\Gamma \vdash \mu X.P} \mu_r \qquad \frac{\Gamma, P(A) \vdash A}{\Gamma, \mu X.P \vdash A} \mu_l \qquad \frac{\Gamma, P(\nu X.P) \vdash B}{\Gamma, \nu X.P \vdash B} \nu_l \qquad \frac{\Gamma, A \vdash P(A)}{\Gamma, A \vdash \nu X.P} \nu_r$
Foncteurs
$\frac{\Gamma, A \vdash B}{\Gamma, P(A) \vdash P(B)} [P] \qquad \frac{\Gamma, A \vdash B}{\Gamma, N(B) \vdash N(A)} [N]$

FIG. 5.1 – Règles de déduction de μLJ

5.1.2 Fragments

On a donné la logique μLJ sous sa forme la plus canonique, et c'est sûrement avec cette généralité qu'elle mérite d'être considérée. Cependant, cette partie vise avant tout à construire un modèle de jeux, et l'expression de ce modèle de jeux est nettement plus simple si on procède à une restriction sur les formules. On va donc introduire ici deux classes particulières de formules, et c'est pour l'intersection des deux qu'on donnera un modèle de jeux.

Notons qu'habituellement, par *fragment* d'une logique, on entend la restriction aux formules engendrées par un ensemble restreint de connecteurs. La propriété fondamentale de cette notion de *fragment* est que c'est un ensemble de formules clos par sous-formule, ce qui paraît être la moindre des choses puisque les formules sont construites inductivement. Ici, on va introduire des fragments de μLJ par restriction sur l'emploi des variables libres et des constructeurs μ et ν : ce ne seront pas des fragments au sens usuel. Par contre, ils partagent avec la notion usuelle de fragment le fait qu'ils engendrent des ensembles stables par sous-formule, propriété qui justifie de les qualifier de *fragments*.

Points fixes stricts. La première restriction est celle aux formules *strictes* : on veut interdire les formules de la forme $\mu X.X$, et de façon générale on ne veut pas de variable libre à *la racine du type* lors d'une construction inductive/coinductive. La raison est que la construction de boucle qui servira en sémantique de jeux à interpréter ce constructeur a besoin d'«espace» : il faut que la sous-arène interprétant la variable libre soit justifiée par un coup interne à l'arène. Bien sûr, tout cela sera revu plus en détails lors de la présentation de la sémantique.

Définition 5.1.6. On définit l'ensemble des variables à la racine d'une formule F par induction :

$$\begin{aligned} \text{rac}(S \Rightarrow T) &= \text{rac}(T) \\ \text{rac}(S + T) &= \emptyset \\ \text{rac}(S \times T) &= \text{rac}(S) \cup \text{rac}(T) \\ \text{rac}(\mu X.T) &= \text{rac}(T) \\ \text{rac}(\nu X.T) &= \text{rac}(T) \\ \text{rac}(X) &= \{X\} \\ \text{rac}(1) &= \emptyset \\ \text{rac}(0) &= \emptyset \end{aligned}$$

On dit alors qu'une formule F est stricte si pour toute sous-formule $\mu X.T$ ou $\nu X.T$ de F , $X \notin \text{rac}(T)$.

Foncteurs pseudo-polynomiaux. Présentons la seconde restriction sur les formules : il s'agit d'imposer qu'aucune variable libre n'apparaisse à droite d'une flèche (bien qu'elles peuvent toujours apparaître à gauche). En effet, la sémantique de μLJ que l'on va présenter s'appuie sur la notion de *foncteur ouvert*, qui transforme en endofoncteur n'importe quelle arène munie de variables libres, à condition qu'aucune de ces variables libres ne justifie d'autre coup. Cette restriction semble arbitraire, mais s'est avérée nécessaire pour rendre la construction des foncteurs ouverts techniquement réalisable.

Définition 5.1.7. Soit F une formule de μLJ . F est pseudo-polynomiale si pour toute sous-formule $S \Rightarrow T$ de F , $\text{fv}(T) = \emptyset$.

Ce fragment est plus petit que nécessaire : on pourrait en effet se restreindre aux formules F telles que, si $S \Rightarrow T$ est une sous-formule de F , alors $rac(T) = \emptyset$. Elles ont bien une interprétation dans le modèle de jeux, mais leur inclusion complique inutilement la structure du modèle catégorique puisqu'il devient nettement moins facile de présenter les formules comme une classe particulière de foncteurs stable par composition.

Remarque. Notons que la classe des formules pseudo-polynomiales / foncteurs pseudo-polynomiaux est une extension directe de celle des formules polynomiales / foncteurs polynomiaux, construits uniquement à partir du produit et de la somme. En pratique, les algèbres initiales et coalgèbres terminales des foncteurs polynomiaux suffisent à modéliser la majorité des types de données utilisés en informatique, par exemple les entiers et les arbres.

Pour modéliser μLJ , on introduira le concept de *catégorie μ -fermée*. Notons que ce concept visera à modéliser μLJ dans sa globalité. On donnera également des définitions affaiblies pour modéliser les fragments stricts et pseudo-polynomiaux de μLJ , définitions affaiblies dont notre catégorie de jeux fournira un exemple concret.

5.2 Réduction

5.2.1 Règles de réduction

La réduction associée aux séquents LJ est usuelle, même si rarement complètement explicitée : les systèmes à base de séquents sont souvent plus du côté «logique» que du côté «calcul» de la correspondance de Curry-Howard, ainsi on montre le Haupsatz [GTL89] en définissant une réduction, mais sans l'explicitier complètement en tant que telle : la stratégie de réduction correspondant à la preuve d'élimination des coupures du calcul des séquents dans [GTL89] n'a pas de sens d'un point de vue programmation, il paraîtrait absurde d'en équiper un langage de programmation. Comme exception notable à cette remarque, mentionnons les travaux dans la lignée de Curien et Herbelin [CH00] étudiant des syntaxes pour les stratégies focalisées sur les séquents.

Ici, on tente de donner des règles de réduction suffisamment générales pour qu'il soit plausible d'interpréter dans μLJ un véritable langage de programmation. Ainsi μLJ peut être vu comme un langage abstrait, sans choix de syntaxe, dans lequel on peut interpréter des langages avec induction (comme le système T) et coinduction pour leur donner une sémantique.

Notation. Comme précédemment, la lettre P désignera des foncteurs positifs et N des foncteurs négatifs. Dans l'exposé des règles qui va suivre, on prendra la liberté de ne pas toujours expliciter l'emploi des règles structurelles, afin de garder les preuves aussi concises que possible. On utilisera l'abréviation :

$$\frac{\Gamma \vdash A}{\Delta \vdash A} S$$

pour désigner l'emploi d'une succession de règles structurelles. On veillera à ne l'utiliser que lorsque les formes de Γ et Δ permettent d'inférer de façon canonique une fonction $\sigma : n \rightarrow p$, où $|\Gamma| = n$ et $|\Delta| = p$, associant à chaque formule de Γ une occurrence de la même formule dans Δ . S'il y a ambiguïté sur cette fonction, on annotera certaines formules par des indices. Il est immédiat que pour toute fonction $\sigma : n \rightarrow p$ telle que Γ_i est toujours une occurrence de

la même formule que $\Delta_{\sigma(i)}$, il existe nécessairement une succession d'applications des règles structurelles implémentant σ .

Cette notation sera utilisée pour les règles de réduction. On suppose donc un choix implicite, pour tous contextes Γ , Δ et $\sigma : |\Gamma| \rightarrow |\Delta|$, d'une telle séquence de règles structurelles.

Coupages logiques. On considère les réductions suivantes sur les coupures entre connecteurs logiques :

(L.1) Coupure $\Rightarrow_r / \Rightarrow_l$

$$\frac{\frac{\frac{\pi_1}{\Gamma, A \vdash B}}{\Gamma \vdash A \Rightarrow B} \Rightarrow_r \quad \frac{\frac{\frac{\pi_2}{\Delta \vdash A} \quad \frac{\pi_3}{\Xi, B \vdash C}}{\Delta, \Xi, A \Rightarrow B \vdash C} \Rightarrow_l}{\Gamma, \Delta, \Xi \vdash C} \text{Cut}}{\Gamma, \Delta, \Xi \vdash C} \text{Cut} \rightsquigarrow \frac{\frac{\frac{\pi_2}{\Delta \vdash A} \quad \frac{\pi_1}{\Gamma, A \vdash B}}{\Delta, \Gamma \vdash B} \text{Cut} \quad \frac{\pi_3}{\Xi, B \vdash C} \text{Cut}}{\frac{\Delta, \Gamma, \Xi \vdash C}{\Gamma, \Delta, \Xi \vdash C} S} \text{Cut}$$

(L.2) Coupure \times_r / \times_l

$$\frac{\frac{\frac{\pi_1}{\Gamma \vdash A} \quad \frac{\pi_2}{\Delta \vdash B}}{\Gamma, \Delta \vdash A \times B} \times_r \quad \frac{\frac{\pi_3}{\Xi, A, B \vdash C}}{\Xi, A \times B \vdash C} \times_l}{\Gamma, \Delta, \Xi \vdash C} \text{Cut} \rightsquigarrow \frac{\frac{\pi_1}{\Gamma \vdash A} \quad \frac{\frac{\pi_2}{\Delta \vdash B} \quad \frac{\pi_3}{\Xi, A, B \vdash C}}{\Delta, \Xi, A \vdash C} \text{Cut}}{\Gamma, \Delta, \Xi \vdash C} \text{Cut}$$

(L.3a) Coupure $\overleftarrow{+}_r / +_l$

$$\frac{\frac{\frac{\pi_1}{\Gamma \vdash A}}{\Gamma \vdash A + B} \overleftarrow{+}_r \quad \frac{\frac{\frac{\pi_2}{\Delta, A \vdash C} \quad \frac{\pi_3}{\Xi, B \vdash C}}{\Delta, \Xi, A + B \vdash C} +_l}{\Gamma, \Delta, \Xi \vdash C} \text{Cut}}{\Gamma, \Delta, \Xi \vdash C} \text{Cut} \rightsquigarrow \frac{\frac{\pi_1}{\Gamma \vdash A} \quad \frac{\pi_2}{\Delta, A \vdash C}}{\Gamma, \Delta \vdash C} \text{Cut}}{\frac{\Gamma, \Delta, \Xi \vdash C}{\Gamma, \Delta, \Xi \vdash C} S} \text{Cut}$$

(L.3b) Coupure $\overrightarrow{+}_r / +_l$

$$\frac{\frac{\frac{\pi_1}{\Gamma \vdash B}}{\Gamma \vdash A + B} \overrightarrow{+}_r \quad \frac{\frac{\frac{\pi_2}{\Delta, A \vdash C} \quad \frac{\pi_3}{\Xi, B \vdash C}}{\Delta, \Xi, A + B \vdash C} +_l}{\Gamma, \Delta, \Xi \vdash C} \text{Cut}}{\Gamma, \Delta, \Xi \vdash C} \text{Cut} \rightsquigarrow \frac{\frac{\pi_1}{\Gamma \vdash B} \quad \frac{\pi_3}{\Xi, B \vdash C}}{\Gamma, \Xi \vdash C} \text{Cut}}{\frac{\Gamma, \Delta, \Xi \vdash C}{\Gamma, \Delta, \Xi \vdash C} S} \text{Cut}$$

(L.4) Coupure $1_r / 1_l$

$$\frac{\frac{\overline{\vdash 1}}{\vdash 1} 1_r \quad \frac{\frac{\pi}{\Gamma \vdash A}}{\Gamma, 1 \vdash A} 1_l}{\Gamma \vdash A} \text{Cut} \rightsquigarrow \frac{\pi}{\Gamma \vdash A}$$

Pas de coupure logique avec 0, puisque celui-ci n'a pas de règle d'introduction à droite.

Points fixes. On considère les réductions suivantes sur les coupures μ_r / μ_l et ν_r / ν_l .

(Fix.1) Coupure μ_r / μ_l

$$\frac{\frac{\frac{\pi_1}{\Gamma \vdash T[\mu X.T/X]}}{\Gamma \vdash \mu X.T} \mu_r \quad \frac{\frac{\pi_2}{\Delta, T[A/X] \vdash A}}{\Delta, \mu X.T \vdash A} \mu_l}{\Gamma, \Delta \vdash A} \text{Cut} \rightsquigarrow$$

$$\frac{\frac{\frac{\pi_1}{\Gamma \vdash T[\mu X.T/X]} \quad \frac{\frac{\frac{\pi_2}{\Delta, T[A/X] \vdash A}}{\Delta, \mu X.T \vdash A} \mu_l}{\Delta, T[\mu X.T/X] \vdash T[A/X]} [T]}{\Gamma, \Delta \vdash T[A/X]} Cut}{\frac{\Gamma, \Delta, \Delta \vdash A}{\Gamma, \Delta \vdash A} S} Cut$$

(Fix.2) Coupure ν_l/ν_r

$$\frac{\frac{\frac{\pi_1}{\Gamma, A \vdash T[A/X]} \quad \frac{\frac{\pi_2}{\Delta, T[\nu X.T/X] \vdash B}}{\Delta, \nu X.T \vdash B} \nu_l}{\Gamma, A \vdash \nu X.T} \nu_r}{\Gamma, A, \Delta \vdash B} Cut \quad \rightsquigarrow \quad \frac{\frac{\frac{\frac{\pi_1}{\Gamma, A \vdash T[A/X]} \quad \frac{\frac{\pi_2}{\Delta, T[\nu X.T/X] \vdash B}}{\Delta, \nu X.T \vdash B} \nu_l}{\Gamma, T[A/X] \vdash T[\nu X.T/X]} [T]}{\Gamma, \Delta, T[A/X] \vdash B} S}{\Gamma, A, \Gamma, \Delta \vdash B} S}{\Gamma, A, \Delta \vdash B} S$$

Remarque. Cette règle apparaît en fait tout naturellement comme un choix d'orientation de la 2-cellule des diagrammes respectifs d'algèbre initiale et de coalgèbre terminale. Prenons le cas de l'algèbre initiale :

$$\begin{array}{ccc} T(\mu X.T) \xrightarrow{T(f^\dagger)} T(B) & & \\ \chi_T \downarrow \quad \Rightarrow \quad \downarrow f & & \\ \mu X.T \xrightarrow{f^\dagger} B & & \end{array}$$

Cette orientation exprime le fait qu'une coupure entre χ_T et f^\dagger se réduit en une nouvelle coupure entre $T(f^\dagger)$ et f . Mais χ_T est juste le terme effectuant un dépliage du point fixe, ce qui correspond à μ_r dans μLJ . La preuve π_2 joue ici le rôle de f , et π_1 est un morphisme précomposé à la coupure entre χ_T et f^\dagger .

Remarque. On notera l'usage des règles foncteurs dans la réduction des points fixes. Leur utilisation n'est pas indispensable, mais la seule alternative qu'on puisse envisager est d'intégrrer l'expansion complète de ces foncteurs dans la règle de réduction (notons que la réduction des coupures de $\mu MALL$ [BM07] utilise cette variante) ce qui rend la réduction non locale et beaucoup plus difficile à analyser d'un point de vue sémantique.

Foncteurs. On équipe μLJ des réductions suivantes :

(Fonc.1a) Constante positive.

$$\frac{\frac{\pi}{\Gamma, A \vdash B}}{\Gamma, P \vdash P} [P](X \notin FV(P)) \quad \rightsquigarrow \quad \frac{P \vdash P^{ax}}{\Gamma, P \vdash P} S$$

(Fonc.1b) Constante négative.

$$\frac{\pi}{\Gamma, B \vdash A} \frac{\pi}{\Gamma, N \vdash N} [N] (X \notin FV(N)) \quad \rightsquigarrow \quad \frac{\overline{N \vdash N}^{ax}}{\Gamma, N \vdash N} S$$

(Fonc.2) Identité.

$$\frac{\pi}{\Gamma, A \vdash B} \frac{\pi}{\Gamma, A \vdash B} [X] \quad \rightsquigarrow \quad \frac{\pi}{\Gamma, A \vdash B}$$

(Fonc.3a) Si P_1 et P_2 sont des foncteurs positifs, la règle $[P_1 \times P_2]$ se déplie en :

$$\frac{\frac{\frac{\frac{\pi}{\Gamma, A \vdash B} [P_1]}{\Gamma, P_1(A) \vdash P_1(B)} S}{\Gamma, P_1(A), P_2(A) \vdash P_1(B)} S}{\Gamma, P_1(A) \times P_2(A) \vdash P_1(B)} \times_l \quad \frac{\frac{\frac{\frac{\pi}{\Gamma, A \vdash B} [P_2]}{\Gamma, P_2(A) \vdash P_2(B)} S}{\Gamma, P_1(A), P_2(A) \vdash P_2(B)} S}{\Gamma, P_1(A) \times P_2(A) \vdash P_2(B)} \times_l}{\frac{\Gamma, P_1(A) \times P_2(A), \Gamma, P_1(A) \times P_2(A) \vdash P_1(B) \times P_2(B)}{\Gamma, P_1(A) \times P_2(A) \vdash P_1(B) \times P_2(B)} S} \times_r$$

(Fonc.3b) Si N_1 et N_2 sont deux foncteurs négatifs, la règle $[N_1 \times N_2]$ se déplie en :

$$\frac{\frac{\frac{\frac{\pi}{\Gamma, B \vdash A} [N_1]}{\Gamma, N_1(A) \vdash N_1(B)} S}{\Gamma, N_1(A), N_2(A) \vdash N_1(B)} S}{\Gamma, N_1(A) \times N_2(A) \vdash N_1(B)} \times_l \quad \frac{\frac{\frac{\frac{\pi}{\Gamma, B \vdash A} [N_2]}{\Gamma, N_2(A) \vdash N_2(B)} S}{\Gamma, N_1(A), N_2(A) \vdash N_2(B)} S}{\Gamma, N_1(A) \times N_2(A) \vdash N_2(B)} \times_l}{\frac{\Gamma, N_1(A) \times N_2(A), \Gamma, N_1(A) \times N_2(A) \vdash N_1(B) \times N_2(B)}{\Gamma, N_1(A) \times N_2(A) \vdash N_1(B) \times N_2(B)} S} \times_r$$

(Fonc.4a) Si P_1 et P_2 sont deux foncteurs positifs, la règle $[P_1 + P_2]$ se déplie en :

$$\frac{\frac{\frac{\frac{\pi}{\Gamma, A \vdash B} [P_1]}{\Gamma, P_1(A) \vdash P_1(B)} S}{\Gamma, P_1(A) \vdash P_1(B) + P_2(B)} \leftarrow_{\vdash_r} \quad \frac{\frac{\frac{\frac{\pi}{\Gamma, A \vdash B} [P_2]}{\Gamma, P_2(A) \vdash P_2(B)} S}{\Gamma, P_2(A) \vdash P_1(B) + P_2(B)} \rightarrow_{\vdash_r}}{\frac{\Gamma, \Gamma, P_1(A) + P_2(A) \vdash P_1(B) + P_2(B)}{\Gamma, P_1(A) + P_2(A) \vdash P_1(B) + P_2(B)} S} \vdash_l$$

(Fonc.4b) Si N_1 et N_2 sont deux foncteurs négatifs, la règle $[N_1 + N_2]$ se déplie en :

$$\frac{\frac{\frac{\frac{\pi}{\Gamma, B \vdash A} [N_1]}{\Gamma, N_1(A) \vdash N_1(B)} S}{\Gamma, N_1(A) \vdash N_1(B) + N_2(B)} \leftarrow_{\vdash_r} \quad \frac{\frac{\frac{\frac{\pi}{\Gamma, B \vdash A} [N_2]}{\Gamma, N_2(A) \vdash N_2(B)} S}{\Gamma, N_2(A) \vdash N_1(B) + N_2(B)} \rightarrow_{\vdash_r}}{\frac{\Gamma, \Gamma, N_1(A) + N_2(A) \vdash N_1(B) + N_2(B)}{\Gamma, N_1(A) + N_2(A) \vdash N_1(B) + N_2(B)} S} \vdash_l$$

(Fonc.5a) Si N est un foncteur négatif et P un foncteur positif, $[N \Rightarrow P]$ se déplie en :

$$\frac{\frac{\frac{\pi}{\Gamma, A \vdash B}}{\Gamma, N(B) \vdash N(A)} [N] \quad \frac{\frac{\pi}{\Gamma, A \vdash B}}{\Gamma, P(A) \vdash P(B)} [P]}{\frac{\frac{\Gamma, N(B), \Gamma, N(A) \Rightarrow P(A) \vdash P(B)}{\Gamma, N(A) \Rightarrow P(A), N(B) \vdash P(B)} S}{\Gamma, N(A) \Rightarrow P(A) \vdash N(B) \Rightarrow P(B)} \Rightarrow_r} \Rightarrow_l$$

(Fonc.5b) Si N est un foncteur négatif et P un foncteur positif, $[P \Rightarrow N]$ se déplie en :

$$\frac{\frac{\frac{\pi}{\Gamma, B \vdash A}}{\Gamma, P(B) \vdash P(A)} [P] \quad \frac{\frac{\pi}{\Gamma, B \vdash A}}{\Gamma, N(A) \vdash N(B)} [N]}{\frac{\frac{\Gamma, P(B), \Gamma, P(A) \Rightarrow N(A) \vdash N(B)}{\Gamma, P(A) \Rightarrow N(A), P(B) \vdash N(B)} S}{\Gamma, P(A) \Rightarrow N(A) \vdash P(B) \Rightarrow N(B)} \Rightarrow_r} \Rightarrow_l$$

(Fonc.6a) Si P est un foncteur positif en X et Y , $[\mu Y.P]$ se déplie en :

$$\frac{\frac{\frac{\pi}{\Gamma, A \vdash B}}{\Gamma, P(A)[\mu Y.P(B)/Y] \vdash P(B)[\mu Y.P(B)/Y]} [P[\mu Y.P(B)/Y]]}{\frac{\frac{\Gamma, P(A)[\mu Y.P(B)/Y] \vdash \mu Y.P(B)}{\Gamma, \mu Y.P(A) \vdash \mu Y.P(B)} \mu_r} \mu_i$$

(Fonc.6b) Si N est un foncteur positif en Y et négatif en X , $[\mu Y.N]$ se déplie en :

$$\frac{\frac{\frac{\pi}{\Gamma, B \vdash A}}{\Gamma, N(A)[\mu Y.N(B)/Y] \vdash N(B)[\mu Y.N(B)/Y]} [N[\mu Y.N(B)/Y]]}{\frac{\frac{\Gamma, N(A)[\mu Y.N(B)/Y] \vdash \mu Y.N(B)}{\Gamma, \mu Y.N(A) \vdash \mu Y.N(B)} \mu_r} \mu_i$$

(Fonc.7a) Si P est un foncteur positif en X et en Y , $[\nu Y.P]$ se déplie en :

$$\frac{\frac{\frac{\pi}{\Gamma, A \vdash B}}{\Gamma, P(A)[\nu Y.P(A)/Y] \vdash P(B)[\nu Y.P(A)/Y]} [P[\nu Y.P(A)/Y]]}{\frac{\frac{\Gamma, \nu Y.P(A) \vdash P(B)[\nu Y.P(A)/Y]}{\Gamma, \nu Y.P(A) \vdash \nu Y.P(B)} \nu_r} \nu_i$$

(Fonc.7b) Si N est un foncteur positif en Y et négatif en X , $[\nu Y.N]$ se déplie en :

$$\frac{\frac{\frac{\pi}{\Gamma, B \vdash A}}{\Gamma, N(A)[\nu Y.N(A)/Y] \vdash N(B)[\nu Y.N(A)/Y]} [N[\nu Y.N(A)/Y]]}{\frac{\frac{\Gamma, \nu Y.N(A) \vdash N(B)[\nu Y.N(A)/Y]}{\Gamma, \nu Y.N(A) \vdash \nu Y.N(B)} \nu_r} \nu_i$$

Proposition 5.2.1. *Soit π une preuve de μLJ , alors on peut éliminer les règles foncteurs dans π . Plus précisément, l'ensemble des réductions (Fonc) données ci-dessus termine.*

Démonstration. Pour chaque variable X , on définit la mesure suivante sur les formules ouvertes.

$$\begin{aligned}
m_X(A) &= 0 \quad (X \notin fv(A)) \\
m_X(T_1 \Rightarrow T_2) &= \max(m_X(T_1), m_X(T_2)) \\
m_X(T_1 + T_2) &= \max(m_X(T_1), m_X(T_2)) \\
m_X(T_1 \times T_2) &= \max(m_X(T_1), m_X(T_2)) \\
m_X(X) &= 0 \\
m_X(\mu Y.T) &= m_X(T) + 1 \\
m_X(\nu Y.T) &= m_X(T) + 1
\end{aligned}$$

Si T est une formule close, on pose $m(T) = 0$. Si T a une unique variable libre X , on définit alors $m(T) = m_X(T)$. Si T a deux variables libres X et Y et si A est une formule close, on prouve l'inégalité suivante :

$$m_X(T[A/Y]) < m_X(\mu Y.T)$$

en effet, on peut prouver par induction immédiate sur T que lorsque A est clos, on a $m_X(T) = m_X(T[A/Y])$. Alors, on a $m_X(T[A/Y]) = m_X(T) < m_X(T) + 1 = m_X(\mu Y.T)$. On a donc une mesure m sur les foncteurs, qui :

- décroît strictement lors de l'expansion des points fixes, par l'inégalité ci-dessus ;
- ne peut que diminuer pour les autres règles, mais alors c'est la taille de la formule qui décroît strictement.

On considère donc l'ordre lexicographique \prec sur les couples $(m(T), |T|)$ (où $|T|$ est la longueur de la formule T). On a bien la propriété que chaque application d'une règle foncteur se réduira en plusieurs applications de règles strictement plus petites pour cet ordre. On conclut donc en considérant, pour chaque preuve π , le multiensemble de toutes les règles foncteurs dans π . On équipe ces multiensembles de l'ordre multiensemble basé sur \prec : cet ordre diminue strictement pour chaque étape d'expansion des foncteurs dans π . \square

Les autres règles de réduction de μLJ vont être dirigées par la notion de coupure *active*, que l'on définit ici.

Définition 5.2.1. *Considérons la coupure suivante :*

$$\frac{\frac{\pi_1}{\Gamma \vdash A} \quad \frac{\pi_2}{\Delta, A \vdash B}}{\Gamma, \Delta \vdash B} \text{Cut}$$

Cette coupure est dite active si on est dans un des cas suivants :

- La première règle de π_2 est la règle d'introduction gauche du connecteur principal de A ;
- A est l'invariant d'une règle ν_r , c'est-à-dire que la coupure a la forme suivante :

$$\frac{\frac{\pi_1}{\Gamma \vdash A} \quad \frac{\frac{\pi_2}{\Delta, A \vdash T(A)}}{\Delta, A \vdash \nu X.T} \nu_r}{\Gamma, \Delta \vdash B} \text{Cut}$$

- A est l'argument d'une règle foncteur, c'est-à-dire que la coupure a l'une des formes suivantes :

$$\frac{\frac{\pi_1}{\Gamma \vdash P(A)} \quad \frac{\pi_2}{\Delta, A \vdash B}}{\Gamma, \Delta \vdash P(B)} [P] \text{Cut} \quad \frac{\frac{\pi_1}{\Gamma \vdash N(A)} \quad \frac{\pi_2}{\Delta, N(A) \vdash N(B)}}{\Gamma, \Delta \vdash N(B)} [N] \text{Cut}$$

Coupages structurelles et axiomes.

(S.1) Axiome à gauche, si la coupure est active :

$$\frac{\frac{A \vdash A}{\Gamma, A \vdash A} \text{ax} \quad \frac{\pi}{\Gamma, A \vdash B}}{\Gamma, A \vdash B} \text{Cut} \quad \rightsquigarrow \quad \frac{\pi}{\Gamma, A \vdash B}$$

(S.2) Coupure à gauche, si la coupure est active.

$$\frac{\frac{\frac{\pi_1}{\Gamma \vdash B} \quad \frac{\pi_2}{\Delta, B \vdash A}}{\Gamma, \Delta \vdash A} \text{Cut} \quad \frac{\pi_3}{\Xi, A \vdash C}}{\Gamma, \Delta, \Xi \vdash C} \text{Cut} \quad \rightsquigarrow \quad \frac{\frac{\pi_1}{\Gamma \vdash B} \quad \frac{\frac{\pi_2}{\Delta, B \vdash A} \quad \frac{\pi_3}{\Xi, A \vdash C}}{\Delta, B, \Xi \vdash C} \text{Cut}}{\Gamma, \Delta, \Xi \vdash C} S \text{Cut}$$

(S.3) Affaiblissement à gauche, si la coupure est active.

$$\frac{\frac{\frac{\pi_1}{\Gamma, \Delta \vdash B}}{\Gamma, A, \Delta \vdash B} W \quad \frac{\pi_2}{\Xi, B \vdash C}}{\Gamma, A, \Delta, \Xi \vdash C} \text{Cut} \quad \rightsquigarrow \quad \frac{\frac{\pi_1}{\Gamma, \Delta \vdash B} \quad \frac{\pi_2}{\Xi, B \vdash C}}{\Gamma, \Delta, \Xi \vdash C} \text{Cut} W$$

(S.4) Contraction à gauche, si la coupure est active.

$$\frac{\frac{\frac{\pi_1}{\Gamma, A, A, \Delta \vdash B}}{\Gamma, A, \Delta \vdash B} C \quad \frac{\pi_2}{\Xi, B \vdash C}}{\Gamma, A, \Delta, \Xi \vdash C} \text{Cut} \quad \rightsquigarrow \quad \frac{\frac{\pi_1}{\Gamma, A, A, \Delta \vdash B} \quad \frac{\pi_2}{\Xi, B \vdash C}}{\Gamma, A, A, \Delta, \Xi \vdash C} \text{Cut} C$$

(S.5) Échange à gauche, si la coupure est active.

$$\frac{\frac{\frac{\pi_1}{\Gamma, B, A, \Delta \vdash C}}{\Gamma, A, B, \Delta \vdash C} \gamma \quad \frac{\pi_2}{\Xi, C \vdash D}}{\Gamma, A, B, \Delta, \Xi \vdash D} \text{Cut} \quad \rightsquigarrow \quad \frac{\frac{\pi_1}{\Gamma, B, A, \Delta \vdash C} \quad \frac{\pi_2}{\Xi, C \vdash D}}{\Gamma, B, A, \Delta, \Xi \vdash D} \text{Cut} \gamma$$

Substitution. Les cas clés décrits ci-dessus n'arrivent que rarement au cours de la réduction, il faut usuellement passer par de très nombreuses étapes de commutation. Ces commutations peuvent être gérées de façon explicite : on aurait alors un système de réduction très puissant, gérant aussi bien l'affaiblissement que la contraction et l'échange de façon explicite. Le revers de la médaille est que si on n'y prend pas garde, de tels systèmes peuvent perdre la propriété de normalisation [Mel95]. De plus, on s'encombre d'une quantité considérable de bureaucratie, qu'on aimerait éviter ici autant que possible. On choisit donc dans cette thèse de donner un système de réécriture non local, qui gère ces commutations de la même manière qu'est gérée la substitution usuelle en λ -calcul. Imaginons qu'on veuille réduire une coupure :

$$\frac{\frac{\pi_1}{\Gamma \vdash A} \quad \frac{\pi_2}{\Delta, A \vdash B}}{\Gamma, \Delta \vdash B} \text{Cut}$$

S'il s'agit d'une coupure active, on retombera dans la plupart des cas sur une coupure logique, structurelle ou axiome. Si ce n'est pas le cas, il s'agit de faire remonter π_1 le long de π_2 jusqu'à atteindre la règle introduisant A à gauche, afin de retomber sur une coupure active sur A . C'est l'objet de la *substitution* que l'on va définir ici. Soit π_1 une preuve d'un séquent $\Gamma \vdash A$ et π_2 une preuve de $\Delta \vdash B$, avec $A \in \Delta$. On définit $\pi_2[\pi_1/A]$ comme en λ -calcul, par induction sur π_2 . Comme hypothèse d'induction, on garde que $\pi_2[\pi_1/A]$ est une preuve de $\Gamma, \Delta \setminus A \vdash B$, où $\Delta \setminus A$ est Δ d'où on a ôté l'occurrence de A substituée.

- Substitution active : si la coupure correspondante (*i.e.* le terme de droite de l'égalité) est active.

$$\left(\frac{\pi_2}{\Delta, A \vdash B} \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_1}{\Gamma \vdash A} \quad \frac{\pi_2}{\Delta, A \vdash B}}{\Gamma, \Delta \vdash B} \text{Cut}$$

Dans tous les autres cas, on suppose que la coupure correspondante n'est pas active.

- Axiome.

$$\left(\frac{}{A \vdash A} \text{ax} \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\pi_1}{\Gamma \vdash A}$$

- Coupure.
 - Si $A \in \Delta$:

$$\left(\frac{\frac{\pi_2}{\Delta \vdash B} \quad \frac{\pi_3}{\Xi, B \vdash C}}{\Delta, \Xi \vdash C} \text{Cut} \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A \vdash B} \quad \frac{\pi_3}{\Xi, B \vdash C}}{\Gamma, \Delta \setminus A, \Xi \vdash C} \text{Cut}$$

- Si $A \in \Xi$:

$$\left(\frac{\frac{\pi_2}{\Delta \vdash B} \quad \frac{\pi_3}{\Xi, B \vdash C}}{\Delta, \Xi \vdash C} \text{Cut} \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2}{\Delta \vdash B} \quad \frac{\pi_3[\pi_1/A]}{\Gamma, \Xi \setminus A, B \vdash C}}{\frac{\Delta, \Gamma, \Xi \setminus A \vdash C}{\Gamma, \Delta, \Xi \setminus A \vdash C} S} \text{Cut}$$

- Contraction.
 - Substitution sur formule contractée :

$$\left(\frac{\frac{\pi_2}{\Delta, A_1, A_2, \Xi \vdash B}}{\Delta, A, \Xi \vdash B} C \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{(\pi_2[\pi_1/A_1])[\pi_1/A_2]}{\frac{\Gamma, \Gamma, \Delta, \Xi \vdash B}{\Gamma, \Delta, \Xi \vdash B} S} S$$

- Substitution sur formule non contractée :

$$\left(\frac{\frac{\pi_2}{\Delta, B, B, \Xi \vdash C}}{\Delta, B, \Xi \vdash C} C \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A, B, B, \Xi \vdash C}}{\Gamma, \Delta \setminus A, B, \Xi \vdash C} C$$

- Affaiblissement

– Substitution sur formule affaiblie :

$$\left(\frac{\pi_2}{\frac{\Delta, \Xi \vdash B}{\Delta, A, \Xi \vdash B}} W \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\pi_2}{\frac{\Delta, \Xi \vdash B}{\Gamma, \Delta, \Xi \vdash B}} S$$

– Substitution sur formule non affaiblie :

$$\left(\frac{\pi_2}{\frac{\Delta, \Xi \vdash C}{\Delta, B, \Xi \vdash C}} W \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\pi_2[\pi_1/A]}{\frac{\Gamma, \Delta \setminus A, \Xi \vdash C}{\Gamma, \Delta \setminus A, B, \Xi \vdash C}} W$$

– Échange.

– Si A est parmi les formules échangées :

$$\left(\frac{\pi_2}{\frac{\Delta' \vdash B}{\Delta \vdash B}} \gamma \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A \vdash B}$$

– Si A n'est pas parmi les formules échangées :

$$\left(\frac{\pi_2}{\frac{\Delta' \vdash B}{\Delta \vdash B}} \gamma \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\pi_2[\pi_1/A]}{\frac{\Gamma, \Delta' \setminus A \vdash B}{\Gamma, \Delta \setminus A \vdash B}} \gamma$$

– Introduction de la flèche à droite.

$$\left(\frac{\pi_2}{\frac{\Delta, B \vdash C}{\Delta \vdash B \Rightarrow C}} \Rightarrow_r \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\pi_2[\pi_1/A]}{\frac{\Gamma, \Delta \setminus A, B \vdash C}{\Gamma, \Delta \setminus A \vdash B \Rightarrow C}} \Rightarrow_r$$

– Introduction de la flèche à gauche.

– Si $A \in \Delta$:

$$\left(\frac{\pi_2}{\frac{\Delta \vdash B}{\Delta, \Xi, B \Rightarrow C \vdash D}} \frac{\pi_3}{\Xi, C \vdash D} \Rightarrow_l \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\pi_2[\pi_1/A]}{\frac{\Gamma, \Delta \setminus A \vdash B}{\Gamma, \Delta \setminus A, \Xi, B \Rightarrow C \vdash D}} \frac{\pi_3}{\Xi, C \vdash D} \Rightarrow_l$$

– Si $A \in \Xi$:

$$\left(\frac{\pi_2}{\frac{\Delta \vdash B}{\Delta, \Xi, B \Rightarrow C \vdash D}} \frac{\pi_3}{\Xi, C \vdash D} \Rightarrow_l \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\pi_2}{\frac{\Delta \vdash B}{\Delta, \Gamma, \Xi \setminus A, B \Rightarrow C \vdash D}} \frac{\pi_3[\pi_1/A]}{\frac{\Gamma, \Xi \setminus A, C \vdash D}{\Gamma, \Delta, \Xi \setminus A, B \Rightarrow C \vdash D}} \Rightarrow_l S$$

– Introduction de 0 à gauche. Impossible, règle sans contexte.

– Introduction de 1 à gauche.

$$\left(\frac{\pi_2}{\frac{\Delta \vdash B}{\Delta, 1 \vdash B}} 1_l \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\pi_2[\pi_1/A]}{\frac{\Gamma, \Delta \setminus A \vdash B}{\Gamma, \Delta \setminus A, 1 \vdash B}} 1_l$$

– Introduction de 1 à droite. Impossible, règle sans contexte.

– Introduction du produit à droite.

– Si $A \in \Delta$:

$$\left(\frac{\frac{\pi_2}{\Delta \vdash B} \quad \frac{\pi_3}{\Xi \vdash C}}{\Delta, \Xi \vdash B \times C} \times_r \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A \vdash B} \quad \frac{\pi_3}{\Xi \vdash C}}{\Gamma, \Delta \setminus A, \Xi \vdash B \times C} \times_r$$

– Si $A \in \Xi$

$$\left(\frac{\frac{\pi_2}{\Delta \vdash B} \quad \frac{\pi_3}{\Xi \vdash C}}{\Delta, \Xi \vdash B \times C} \times_r \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2}{\Delta \vdash B} \quad \frac{\pi_3[\pi_1/A]}{\Gamma, \Xi \setminus A \vdash C}}{\Delta, \Gamma, \Xi \setminus A \vdash B \times C} \times_r \quad S$$

– Introduction du produit à gauche.

$$\left(\frac{\frac{\pi_2}{\Delta, B, C \vdash D}}{\Delta, B \times C \vdash D} \times_l \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A, B, C \vdash D}}{\Gamma, \Delta \setminus A, B \times C \vdash D} \times_l$$

– Introduction gauche de la somme à droite.

$$\left(\frac{\frac{\pi_2}{\Delta \vdash B}}{\Delta \vdash B + C} \leftarrow_r \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A \vdash B}}{\Gamma, \Delta \setminus A \vdash B + C} \leftarrow_r$$

– Introduction droite de la somme à droite.

$$\left(\frac{\frac{\pi_2}{\Delta \vdash C}}{\Delta \vdash B + C} \rightarrow_r \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A \vdash C}}{\Gamma, \Delta \setminus A \vdash B + C} \rightarrow_r$$

– Introduction de la somme à gauche.

– Si $A \in \Delta$:

$$\left(\frac{\frac{\pi_2}{\Delta, B \vdash D} \quad \frac{\pi_3}{\Xi, C \vdash D}}{\Delta, \Xi, B + C \vdash D} +_l \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A, B \vdash D} \quad \frac{\pi_3}{\Xi, C \vdash D}}{\Gamma, \Delta \setminus A, \Xi, B + C \vdash D} +_l$$

– Si $A \in \Xi$

$$\left(\frac{\frac{\pi_2}{\Delta, B \vdash D} \quad \frac{\pi_3}{\Xi, C \vdash D}}{\Delta, \Xi, B + C \vdash D} +_l \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2}{\Delta, B \vdash D} \quad \frac{\pi_3[\pi_1/A]}{\Gamma, \Xi \setminus A, C \vdash D}}{\Delta, \Gamma, \Xi \setminus A, B + C \vdash D} +_l \quad S$$

– Introduction de μ à droite.

$$\left(\frac{\frac{\pi_2}{\Delta \vdash T(\mu X.T)}}{\Delta \vdash \mu X.T} \mu_r \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A \vdash T(\mu X.T)}}{\Gamma, \Delta \setminus A \vdash \mu X.T} \mu_r$$

– Introduction de μ à gauche.

$$\left(\frac{\frac{\pi_2}{\Delta, T(B) \vdash B}}{\Delta, \mu X.T \vdash B} \mu_l \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A, T(B) \vdash B}}{\Gamma, \Delta \setminus A, \mu X.T \vdash B} \mu_l$$

– Introduction de ν à gauche.

$$\left(\frac{\frac{\pi_2}{\Delta, T(\nu X.T) \vdash B}}{\Delta \nu X.T \vdash B} \nu_l \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A, T(\nu X.T) \vdash B}}{\Gamma, \Delta \setminus A, \nu X.T \vdash B} \nu_l$$

– Introduction de ν à droite.

$$\left(\frac{\frac{\pi_2}{\Delta, B \vdash T(B)}}{\Delta, B \vdash \nu X.T} \nu_r \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A, B \vdash T(B)}}{\Gamma, \Delta \setminus A, B \vdash \nu X.T} \nu_r$$

– Foncteur positif.

$$\left(\frac{\frac{\pi_2}{\Delta, B \vdash C}}{\Delta, P(B) \vdash P(C)} [P] \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A, B \vdash C}}{\Gamma, \Delta \setminus A, P(B) \vdash P(C)} [P]$$

– Foncteur négatif.

$$\left(\frac{\frac{\pi_2}{\Delta, C \vdash B}}{\Delta, N(B) \vdash N(C)} [N] \right) \left[\frac{\pi_1}{\Gamma \vdash A} / A \right] = \frac{\frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \setminus A, C \vdash B}}{\Gamma, \Delta \setminus A, N(B) \vdash N(C)} [N]$$

À l'aide de cette notion de substitution, on peut finaliser la réduction sur μLJ , en ajoutant la règle suivante :

(Sub) Si la coupure suivante est non active :

$$\frac{\frac{\pi_1}{\Gamma \vdash A} \quad \frac{\pi_2}{\Delta, A \vdash B}}{\Gamma, \Delta \vdash B} Cut \rightsquigarrow \frac{\pi_2[\pi_1/A]}{\Gamma, \Delta \vdash B}$$

Réduction et règles affaiblies. La proposition suivante permet de considérer les règles affaiblies comme également équipées de cette réduction.

Proposition 5.2.2. *La logique μLJ affaiblie est stable par réduction \rightsquigarrow .*

Démonstration. Aucune règle de réduction ne fait apparaître d'applications de μ_l, μ_r, ν_l ou ν_r (bien sur, certaines réductions peuvent les dupliquer), ainsi si π ne contient que des applications de ces règles sans contexte, ce sera également le cas de π' . De même, les règles foncteurs n'apparaissent que dans les réductions (**Fix**), et elles dépendent d'un contexte si et seulement si les règles μ_l et ν_r réduites dépendaient d'un contexte. \square

5.2.2 Coupures irréductibles

Le système de réduction présenté ci-dessus, bien qu'assez complexe, n'est pas complet. Exposons ici les coupures qu'il ne réduit pas, et justifions les choix de conception qui ont amené à ces absences. Pour qu'une coupure ne soit pas réductible, il faut tout d'abord qu'elle soit active, sinon on lancerait la substitution. Examinons donc les différentes coupures actives possibles. Partons d'une coupure :

$$\frac{\frac{\pi_1}{\Gamma \vdash A} \quad \frac{\pi_2}{\Delta, A \vdash B}}{\Gamma, \Delta \vdash B} \text{Cut}$$

Si π_2 commence par une règle d'introduction à gauche. Plusieurs sous-cas. Si π_1 commence par une règle d'introduction à droite, la coupure est nécessairement une coupure logique et se réduit sans encombre. Si π_1 commence par une règle structurelle ou axiome, la coupure est nécessairement une coupure structurelle ou axiome et se réduit sans encombre. Si π_1 commence par une règle d'introduction à gauche, **pas de réduction.** Cependant, ce cas n'arrive pas dans la traduction d'un λ -calcul (comme on le verra formellement avec le système T), et ne bloquera jamais une réduction de tête, réduction que simulera la sémantique de jeux. On pourrait aisément rajouter certaines réductions et les valider par la sémantique mais on préfère éviter de surcharger la réduction, déjà bien assez complexe. De plus, ce cas comprend la coupure suivante :

$$\frac{\frac{\pi_1}{\Gamma, T(A) \vdash A} \quad \frac{\pi_2}{\Delta, A \vdash B}}{\Gamma, \mu X.T, \Delta \vdash B} \text{Cut}$$

Or, il n'y a aucun espoir de réduire cette coupure en préservant la normalisation. En effet, la seule solution pour que π_2 puisse remonter dans π_1 et rencontrer la règle d'introduction de A serait d'autoriser l'expansion suivante :

$$\frac{\pi}{\Gamma, T(A) \vdash A} \quad \frac{\pi}{\Gamma, \mu X.T \vdash A} \mu_l \quad \rightsquigarrow \quad \frac{\frac{\frac{\pi}{\Gamma, T(A) \vdash A} \quad \frac{\pi}{\Gamma, \mu X.T \vdash A} \mu_l}{\Gamma, T(\mu X.T) \vdash T(A)} [T] \quad \frac{\pi}{\Gamma, T(A) \vdash A}}{\Gamma, \mu X.T \vdash T(A)} \text{co}\mu_l \quad \frac{\pi}{\Gamma, T(A) \vdash A}}{\frac{\Gamma, \mu X.T, \Gamma \vdash A}{\Gamma, \mu X.T \vdash A} S} \text{Cut}$$

Où $\text{co}\mu_l$ est la règle inverse de μ_r , dérivable dans μLJ par la Proposition 1.3.1. Cette expansion est validée par la sémantique : elle correspond très précisément à une 2-cellule dans le diagramme d'algèbre initiale. Ainsi, on pourrait avoir une interaction entre π_1 et π_2 , au prix d'une évidente rupture de la normalisation. Dans le cadre de $\mu MALL$ [BM07], cette difficulté est contournée en intégrant cette coupure problématique dans la règle μ_l , ce qui correspondrait dans μLJ à remplacer la règle μ_l par la variante suivante :

$$\frac{\Gamma, T(A) \vdash A \quad \Delta, A \vdash B}{\Gamma, \Delta, \mu X.T \vdash B} \mu'_l$$

Ainsi, on peut récupérer un résultat d'élimination des «coupures». Bien sûr, une preuve sans «coupure» dans un tel système ne satisferait pas la propriété de la sous-formule ni aucune des propriétés structurelles usuelles des preuves sans coupures, ce qui en limite l'intérêt ici.

Si π_2 commence par une autre règle. La contrainte d'activité de la coupure ne laisse que deux cas possibles. Le premier est celui où π_2 commence par une règle foncteur. Dans ce cas, il est toujours possible d'éliminer la règle foncteur (on a vu qu'on pouvait le faire,

voir la Proposition 5.2.1) et de se ramener à un des autres cas. Finalement, reste le cas où π_2 commence par la règle ν_r , et où la formule A coupée sert d'invariant pour la définition coinductive. Autrement dit, on est dans le cas suivant :

$$\frac{\frac{\pi_1}{\Gamma \vdash A} \quad \frac{\frac{\pi_2}{\Delta, A \vdash T(A)}}{\Delta, A \vdash \nu X.T} \nu_r}{\Gamma, \Delta \vdash B} Cut$$

Dans ce cas, **la réduction est également bloquée**. Cependant, ce cas est dual du cas précédemment explicité pour la règle μ_l , on a donc pas d'espoir de débloquent la coupure tant que le connecteur ν à droite n'a pas rencontré un ν à gauche. L'intuition ici est celle de la *paresse* : il est bien connu en informatique que manipuler des structures coinductives avec une évaluation stricte mène à la divergence.

Normalisation. Même si ces règles ont été conçues pour donner une réduction qui termine, on ne le prouvera pas dans cette thèse. On défend en fait l'idée que l'interprétation à venir de ces séquents par des stratégies totales vaut aussi bien qu'un résultat syntaxique de normalisation, pour plusieurs raisons :

La première est que comme on le verra en Section 8.3, la définissabilité sur les stratégies obtenues permet d'obtenir un résultat d'élimination des coupures dans μLJ au prix d'un passage dans une syntaxe infinitaire, ce qu'aucune réduction locale sur la syntaxe finitaire de μLJ ne pourrait permettre. En particulier, on en déduira un algorithme de normalisation par évanouissement pour les preuves d'un séquent $\Gamma \vdash A$ sans occurrences de μ ou de ν .

Enfin, rappelons que dans la partie précédente de cette thèse on s'est appuyés sur le fait que l'interaction entre stratégies innocentes entretient des liens très forts avec la réduction linéaire de tête dans la syntaxe [DHR96b], ainsi la terminaison des interactions entre stratégies du modèle de jeux de μLJ revient à un résultat de normalisation d'une notion (non définie explicitement) de réduction linéaire de tête sur les preuves de μLJ .

5.3 Expressivité

Pour démontrer l'expressivité de cette logique et pour illustrer son aspect dynamique, on donne une traduction du système T de Gödel dans μLJ et on montre qu'elle est valide. On rappelle succinctement la syntaxe des termes et des types, une étude plus complète peut être trouvée par exemple dans [GTL89].

5.3.1 Quotient structurel

Le système T et μLJ gèrent le contexte de manière totalement différente : là où μLJ garde un contrôle très explicite sur son contexte et sur l'ordre et la multiplicité des formules qui y apparaissent, le système T ne s'occupe que des noms de variables qui apparaissent dans le contexte. Ainsi, on peut à n'importe quel endroit du terme utiliser une variable quelle que soit sa position dans le contexte : c'est loin d'être le cas dans μLJ , où il convient de procéder à une succession d'affaiblissements, afin qu'il reste juste dans le contexte une unique occurrence de la formule que l'on veut utiliser.

Bien sûr, ce genre de distinction amènera quelques subtilités lors de la traduction du système T dans μLJ . En particulier, il faudra introduire toute une couche de gestion du contexte,

et il se trouve que la simulation du système T dans μLJ ne sera valide qu'à remaniement de cette couche structurelle près.

Définition 5.3.1. *On considère la relation d'équivalence structurelle \simeq sur les preuves, définie comme la fermeture réflexive, transitive, symétrique et contextuelle des deux relations suivantes :*

- Si les séquences structurelles S et S' dans les deux preuves ci-dessous engendrent la même fonction $\sigma : |\Delta| \rightarrow |\Gamma|$, alors :

$$\frac{\pi}{\frac{\Delta \vdash A}{\Gamma \vdash A}} S \simeq \frac{\pi}{\frac{\Delta \vdash A}{\Gamma \vdash A}} S'$$

- Si π peut être obtenue de π' par commutation entre une règle logique et une règle structurelle agissant sur des formules différentes, alors $\pi \simeq \pi'$.

Le quotient est décidable : il suffit de faire descendre le plus possible les règles structurelles dans la preuve, puis de comparer les fonctions engendrées sur les contextes. De plus, on peut vérifier que \simeq est déjà contenue dans la fermeture réflexive, transitive et symétrique de la réduction \rightsquigarrow , ainsi cette équivalence ne rajoute rien à la théorie équationnelle de μLJ . Notons qu'on ne le vérifiera pas formellement ici, pour pouvoir se concentrer sur l'étude de μLJ .

5.3.2 Rappel : définition du système T

Définition 5.3.2. *Les types du système T sont donnés par la syntaxe suivante :*

$$A, B ::= \text{int} \mid \text{bool} \mid A \Rightarrow B \mid A \times B$$

Les termes sont donnés par la grammaire :

$$M, N, P ::= x \mid \lambda x.M \mid MN \mid 0 \mid S \mid \text{tt} \mid \text{ff} \mid \text{if } M \ N \ P \mid \langle M, N \rangle \mid \pi^1 M \mid \pi^2 M \mid \text{rec } M \ N \ P$$

On considère les règles usuelles du λ -calcul simplement typé avec produits :

$$\frac{}{\Gamma \vdash x : A} (x : A) \in \Gamma \quad \frac{\Gamma \vdash M : B}{\Gamma \setminus (x : A) \vdash \lambda x.M : A \Rightarrow B} (x : A) \in \Gamma$$

$$\frac{\Gamma \vdash M : A \Rightarrow B \quad \Gamma \vdash N : A}{\Gamma \vdash (MN) : B} \quad \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \langle M, N \rangle : A \times B}$$

$$\frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \pi^1 M : A} \quad \frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \pi^2 M : B}$$

Auxquelles on ajoute les règles suivantes :

$$\frac{}{\Gamma \vdash 0 : \text{int}} \quad \frac{}{\Gamma \vdash \text{tt} : \text{bool}} \quad \frac{}{\Gamma \vdash \text{ff} : \text{bool}}$$

$$\frac{\Gamma \vdash M : \text{int}}{\Gamma \vdash S \ M : \text{int}} \quad \frac{\Gamma \vdash B : \text{bool} \quad \Gamma \vdash M : A \quad \Gamma \vdash N : A}{\Gamma \vdash \text{if } B \ M \ N : A}$$

$$\frac{\Gamma \vdash T : A \quad \Gamma \vdash M : A \Rightarrow A \quad \Gamma \vdash N : \text{int}}{\Gamma \vdash \text{rec } T \ M \ N : A}$$

Définition 5.3.3. On définit la réduction \rightsquigarrow^T sur les termes du système T comme l'union des réductions suivantes :

$$\begin{aligned}
(\lambda x.M)N &\rightsquigarrow^T N[M/x] \\
\pi^1 \langle M, N \rangle &\rightsquigarrow^T M \\
\pi^2 \langle M, N \rangle &\rightsquigarrow^T N \\
\text{if tt } M N &\rightsquigarrow^T M \\
\text{if ff } M N &\rightsquigarrow^T N \\
\text{rec } T M 0 &\rightsquigarrow^T T \\
\text{rec } T M (S N) &\rightsquigarrow^T M (\text{rec } T M N)
\end{aligned}$$

On peut prouver pour cette réduction non typée le résultat usuel de *préservation du sujet*, qui montre que le typage est préservé par réduction. Ainsi, la réduction ci-dessus peut être vue comme une réduction sur les arbres de preuve, et c'est en fait celle-ci qui sera interprétée dans μLJ . Notons que l'extension des règles ci-dessus à une réduction sur les arbres de preuve est directe, à un détail près : on fixe la convention que si on a $\Delta \vdash M : B$ avec $(x : A) \in \Delta$ et $\Gamma \vdash N : A$, alors $\Gamma, \Delta \setminus (x : A) \vdash M[N/x] : B$, c'est-à-dire que l'opération de substitution place le contexte correspondant à N en premier dans le contexte du terme obtenu, comme le fait le mécanisme de substitution dans μLJ .

5.3.3 Traduction dans μLJ

Définition 5.3.4. A chaque type A du système T , on va associer une formule A^* de μLJ comme suit :

$$\begin{aligned}
\text{int}^* &= \mu X.1 + X \\
\text{bool}^* &= 1 + 1 \\
(T_1 \Rightarrow T_2)^* &= T_1^* \Rightarrow T_2^* \\
(T_1 \times T_2)^* &= T_1^* \times T_2^*
\end{aligned}$$

A un contexte Γ on associe un contexte Γ^* de μLJ en appliquant élément par élément la traduction ci-dessus.

Définition 5.3.5. A chaque terme typé $\Gamma \vdash M : A$ du système T , associons une preuve M^* de $\Gamma^* \vdash A^*$ comme suit. On prendra les abréviations $\text{nat} = \mu X.1 + X$ et $\text{bool} = 1 + 1$. Soit M un terme du système T dont les variables libres sont incluses dans Γ , on définit sa traduction dans le contexte Γ , désignée par M^* (en laissant Γ implicite), comme suit :

- Variable, si x est introduit dans un contexte Γ :

$$x^* = \frac{\overline{ax}}{\overline{\Gamma \vdash A}} S$$

- Abstraction.

$$(\lambda x.M)^* = \frac{\overline{M^*}}{\overline{\Gamma \vdash B}} S \quad \frac{\overline{\Gamma \setminus A, A \vdash B}}{\overline{\Gamma \setminus A \vdash A \Rightarrow B}} \Rightarrow_r$$

- *Application.*

$$(MN)^* = \frac{\frac{N^*}{\Gamma \vdash A} \quad \frac{\frac{M^*}{\Gamma \vdash A \Rightarrow B} \quad \frac{\frac{\overline{A \vdash A} \quad \overline{B \vdash B}}{A, A \Rightarrow B \vdash B} \Rightarrow_l}{\Gamma, A \vdash B} Cut}{\frac{\Gamma, \Gamma \vdash B}{\Gamma \vdash B} S} Cut$$

- *Paire.*

$$\langle M, N \rangle^* = \frac{\frac{M^*}{\Gamma \vdash A} \quad \frac{N^*}{\Gamma \vdash B}}{\frac{\Gamma, \Gamma \vdash A \times B}{\Gamma \vdash A \times B} S} \times_r$$

- *Projection gauche.*

$$(\pi^1 M)^* = \frac{\frac{M^*}{\Gamma \vdash A \times B} \quad \frac{\frac{\overline{A \vdash A} \quad \overline{B \vdash A}}{A, B \vdash A} W}{A \times B \vdash A} \times_l}{\Gamma \vdash A} Cut$$

- *Projection droite.*

$$(\pi^2 M)^* = \frac{\frac{M^*}{\Gamma \vdash A \times B} \quad \frac{\frac{\overline{B \vdash B} \quad \overline{A \vdash B}}{A, B \vdash B} W}{A \times B \vdash B} \times_l}{\Gamma \vdash B} Cut$$

- *Vrai.*

$$\mathbf{tt}^* = \frac{\frac{\overline{\vdash 1} \quad \overline{1_r}}{\vdash \mathbf{bool}} \leftarrow_r}{\Gamma \vdash \mathbf{bool}} S$$

- *Faux.*

$$\mathbf{ff}^* = \frac{\frac{\overline{\vdash 1} \quad \overline{1_r}}{\vdash \mathbf{bool}} \rightarrow_r}{\Gamma \vdash \mathbf{bool}} S$$

- *Conditionnelle.*

$$(\mathbf{if} B M N)^* = \frac{\frac{B^*}{\Gamma \vdash \mathbf{bool}} \quad \frac{\frac{M^*}{\Gamma \vdash A} \quad \frac{N^*}{\Gamma \vdash A}}{\Gamma, 1 \vdash A} 1_l \quad \frac{\overline{\Gamma \vdash A}}{\Gamma, 1 \vdash A} 1_l}{\frac{\Gamma, \Gamma, \mathbf{bool} \vdash A}{\Gamma, \Gamma, \Gamma \vdash A} Cut} \times_l$$

- *Zero.*

$$0^* = \frac{\frac{\overline{\vdash 1} 1_r}{\vdash 1} \leftarrow_r}{\frac{\vdash \text{nat}}{\Gamma \vdash \text{nat}}} \mu_r S$$

- *Successeur.*

$$(SN)^* = \frac{\frac{N^*}{\Gamma \vdash \text{int}} \rightarrow_r}{\frac{\Gamma \vdash 1 + \text{int}}{\Gamma \vdash \text{int}}} \mu_r$$

- *Récursur.*

$$(\text{rec } T \ M \ N)^* = \frac{\frac{N^*}{\Gamma \vdash \text{int}} \quad \frac{\frac{\frac{T^*}{\Gamma \vdash A} 1_l \quad \frac{\frac{M^*}{\Gamma \vdash A \Rightarrow A} \quad \frac{\overline{A \vdash A} \ ax \quad \overline{A \vdash A} \ ax}{A, A \Rightarrow A \vdash A} \Rightarrow_l}{\Gamma, A \vdash A} \text{Cut}}{\Gamma, \Gamma, 1 + A \vdash A} +_l}{\frac{\Gamma, \Gamma, \text{int} \vdash A}{\Gamma, \Gamma, \Gamma \vdash A} \mu_i \text{Cut}} S$$

Avant de montrer qu'on a bien une simulation, il nous faut montrer comment mettre en relation les opérations de substitution de ces deux calculs :

Lemme 5.3.1. *Soient deux termes $\Delta \vdash M : B$ avec $(x : A) \in \Delta$ et $\Gamma \vdash N : A$. Alors :*

$$(M[N/x])^* \simeq M^*[N^*/A]$$

Démonstration. Par induction sur M , chaque cas est obtenu par un calcul direct. \square

Théorème 5.3.2. *Cette traduction est une simulation à \simeq près, i.e. si $M \rightsquigarrow^T N$, alors $M^* \rightsquigarrow^* N'$ où $N' \simeq N^*$.*

Démonstration. On considère les règles de réduction une à une. Le cas de la β -réduction est une conséquence directe du Lemme 5.3.1, les autres cas se vérifient directement, par un calcul fastidieux mais direct. \square

Étant donné que l'équivalence \simeq est incluse dans la relation d'équivalence générée par \rightsquigarrow , on peut immédiatement déduire de ce théorème que tout modèle de μLJ est un modèle du système T . En particulier, on donnera plus tard un modèle de jeux, où chaque preuve de μLJ sera interprétée par une stratégie totale ; donc par le théorème ci-dessus on sera également capable d'associer une stratégie totale à chaque terme du système T .

Chapitre 6

Sémantique catégorique de μLJ

Sommaire

6.1	Catégories faiblement μ-fermées	153
6.2	Relèvement à \mathcal{C}_Γ	155
6.2.1	Foncteurs forts	155
6.2.2	Cas contravariant	157
6.2.3	Relèvement des points fixes	159
6.2.4	Uniformité du relèvement	160
6.2.5	Discussion : construction des forces	164
6.3	Modélisation de μLJ	166
6.3.1	Catégories μ -fermées	166
6.3.2	Lemmes de substitution	167
6.4	Interprétation	168
6.4.1	Interprétation d'une preuve	168
6.4.2	Correction de l'interprétation	171

Dans ce chapitre, on va introduire progressivement la structure nécessaire à l'interprétation correcte de μLJ . Pour comprendre la structure catégorique nécessaire à la modélisation de μLJ , on traitera tout d'abord le cas particulier de la version *faible* de μLJ , c'est-à-dire restreinte aux règles *affaiblies* présentées en Section 5.1.1. Ensuite, on donnera une structure de foncteurs *forts*, qui est nécessaire pour interpréter les règles générales de μLJ de façon compatible avec leur réduction.

6.1 Catégories faiblement μ -fermées

Si on s'intéresse uniquement à la modélisation de la forme affaiblie de μLJ , on peut définir l'interprétation exactement dans les mêmes lignes que pour le μ -calcul propositionnel dans les catégories μ -bicomplètes [San02b].

Interprétation partielle. On suppose que la catégorie de destination \mathcal{C} est cartésienne fermée et a les coproduits faibles fonctoriels : c'est un modèle¹ de LJ . Si \mathcal{X} est un ensemble fini

¹A condition, bien sûr, que notre version de LJ ne comprenne pas la règle de commutation correspondant à l'équation $[f|g]; h = [f; h|g; h]$. Sinon il faudrait que \mathcal{C} soit bicartésienne fermée, et comme justifié dans la Section 2.2.6 on aurait alors peu d'espoir d'en donner un modèle à base de jeux.

de variables libres, désignons par $\mathcal{C}^{\mathcal{X}}$ le produit de $|\mathcal{X}|$ copies de \mathcal{C} . Alors il est habituel d'interpréter une formule F de LJ avec des variables libres choisies dans \mathcal{X} par un endofoncteur :

$$\llbracket F \rrbracket : (\mathcal{C}^{op} \times \mathcal{C})^{\mathcal{X}} \rightarrow \mathcal{C}$$

Implicitement, cette interprétation dépend du contexte \mathcal{X} choisi. Formellement, il faudrait donc écrire $\llbracket F \rrbracket_{\mathcal{X}}$ mais on s'en passera la plupart du temps, lorsque le contexte ne sera pas ambigu. Si F est une formule close, alors on assimilera le foncteur constant $\llbracket F \rrbracket_{\emptyset}$ et son objet image $\llbracket F \rrbracket_{\emptyset}() \in \mathcal{C}_0$. Notons que les foncteurs ne sont que des intermédiaires. Au final, on s'intéressera toujours à l'interprétation de formules closes puisque c'est sur elles que les règles de la logique sont définies.

Cette interprétation des formules s'étend en une interprétation partielle des formules à variables libres de μLJ dans \mathcal{C} , de la manière qui suit.

Définition 6.1.1 (Interprétation partielle). *Soit \mathcal{C} cartésienne fermée munie de coproduits faibles fonctoriels. On définit par induction l'interprétation des formules de μLJ comme suit :*

- $\llbracket 0 \rrbracket_{\mathcal{X}}$ est le foncteur constant égal à 0, où 0 est l'objet faiblement initial de \mathcal{C} ;
- $\llbracket 1 \rrbracket_{\mathcal{X}}$ est le foncteur constant égal à 1, où 1 est l'objet terminal de \mathcal{C} ;
- $\llbracket S \times T \rrbracket_{\mathcal{X}} = \llbracket S \rrbracket_{\mathcal{X}} \times \llbracket T \rrbracket_{\mathcal{X}}$;
- $\llbracket S + T \rrbracket_{\mathcal{X}} = \llbracket S \rrbracket_{\mathcal{X}} + \llbracket T \rrbracket_{\mathcal{X}}$;
- $\llbracket S \Rightarrow T \rrbracket_{\mathcal{X}} = \llbracket S \rrbracket_{\mathcal{X}} \Rightarrow \llbracket T \rrbracket_{\mathcal{X}}$;
- $\llbracket \mu X.T \rrbracket_{\mathcal{X} \setminus \{X\}}$ est, si elle existe, l'algèbre initiale paramétrée de $\llbracket T \rrbracket_{\mathcal{X}}$.
- $\llbracket \nu X.T \rrbracket_{\mathcal{X} \setminus \{X\}}$ est, si elle existe, la coalgèbre terminale paramétrée de $\llbracket T \rrbracket_{\mathcal{X}}$.

Notons que dans ces deux derniers cas, la formule $\mu X.T$ n'est valide que lorsque $X \notin fv^-(T)$, aussi le domaine de $\llbracket T \rrbracket_{\mathcal{X}}$ peut se mettre sous la forme $\mathcal{C} \times (\mathcal{C}^{op} \times \mathcal{C})^{\mathcal{X} \setminus \{X\}}$, et c'est vis-à-vis de cette forme qu'on prend l'éventuelle algèbre initiale paramétrée ou coalgèbre terminale paramétrée.

Catégories faiblement μ -fermées. Les catégories faiblement μ -fermées sont alors données par la définition suivante :

Définition 6.1.2 (Catégorie faiblement μ -fermée). *Soit \mathcal{C} une catégorie cartésienne fermée et munie de coproduits faibles fonctoriels. \mathcal{C} est faiblement μ -fermée si $\llbracket - \rrbracket$ est une fonction totale.*

Autrement dit, \mathcal{C} est faiblement μ -fermée lorsqu'elle possède suffisamment d'algèbres initiales et coalgèbres finales pour interpréter les formules de μLJ . On a la Proposition suivante :

Proposition 6.1.1. *On peut étendre la fonction $\llbracket - \rrbracket$ en une interprétation sur les preuves invariante par réduction. Autrement dit, toute catégorie faiblement μ -fermée est un modèle des règles affaiblies de μLJ .*

Démonstration. On ne prouve pas cette proposition, mais elle découlera immédiatement du théorème sur les catégories μ -fermées générales. \square

Exemple. La catégorie \mathcal{E} modèle de FPC dans [McC00] est faiblement μ -fermée. En effet, les invariants minimaux (introduits informellement en Section 2.4.1) sont en particulier des algèbres initiales et coalgèbres terminales. En particulier, n'importe quelle catégorie pour laquelle les foncteurs syntaxiques sont équipés d'invariants minimaux sera faiblement μ -fermée,

c'est le cas de nombreuses catégories de domaines. Tous ces modèles sont dégénérés, au sens qu'ils assimilent algèbres initiales et coalgèbres terminales : μ et ν ont la même interprétation. Dans cette partie, on construira un modèle de jeux non dégénéré en assurant la distinction des algèbres initiales et coalgèbres terminales via des conditions de gain.

Caractère non ensembliste. Notons en passant que **Ens** n'est *pas* faiblement μ -fermée. En effet, posons $2 = 1 + 1$. Toute catégorie faiblement μ -fermée possède une interprétation pour la formule $\mu X.(X \Rightarrow 2) \Rightarrow 2$. Dans **Ens**, cette interprétation serait un ensemble E vérifiant l'isomorphisme $(E \Rightarrow \{0, 1\}) \Rightarrow \{0, 1\} \cong E$, autrement dit :

$$\mathcal{P}(\mathcal{P}(E)) \cong E$$

ce qui est impossible pour des raisons évidentes de cardinalité. C'est pour cette raison que de nombreux auteurs se restreignent aux formules *strictement positives* [CD94, AA99], c'est-à-dire qu'aucune variable libre n'apparaît à gauche d'une flèche. On se passe ici de cette restriction puisque le modèle de jeux s'accommode sans mal des formules non strictement positives.

6.2 Relèvement à \mathcal{C}_Γ

Rappelons que dans les préliminaires, on a introduit \mathcal{C}_Γ comme la catégorie de co-Kleisli de la comonade $\Gamma \times -$: intuitivement, c'est la catégorie dont les morphismes ont des «variables libres» dans Γ . Cette construction est nécessaire et habituelle pour interpréter le λ -calcul dans une catégorie cartésienne fermée, voir [LS88]. On s'intéresse dans cette section à l'interprétation des règles générales de μLJ . Comme on va le voir, celles-ci nécessitent un relèvement des foncteurs sur \mathcal{C}_Γ .

6.2.1 Foncteurs forts

Pour simplifier la discussion dans cette sous-section, restreignons-nous dans un premier temps aux foncteurs covariants. On aimerait introduire ici les bases de l'interprétation de la règle suivante, où P est une formule de μLJ avec une unique variable libre $X \in fv^+(P)$.

$$\frac{\Gamma, A \vdash B}{\Gamma, P(A) \vdash P(B)} [P]$$

Nécessité des relèvements. On a vu dans la section précédente que dans une catégorie faiblement μ -fermée \mathcal{C} , toute formule P vérifiant les conditions ci-dessus peut être interprétée par un foncteur $\llbracket P \rrbracket : \mathcal{C} \rightarrow \mathcal{C}$. Il est alors immédiat de se rendre compte qu'interpréter la règle ci-dessus correspond à exiger un *relèvement* de $\llbracket P \rrbracket$ sur $\mathcal{C}_\Gamma \rightarrow \mathcal{C}_\Gamma$. Par la Proposition 1.3.5, cela revient également à exiger que $\llbracket P \rrbracket$ soit équipé d'une loi distributive :

$$\lambda : (\Gamma \times -)\llbracket P \rrbracket \rightarrow \llbracket P \rrbracket(\Gamma \times -)$$

Forces. De telles lois distributives $\theta_A^T : \Gamma \times T(A) \rightarrow T(\Gamma \times A)$ sont en général qualifiées de *forces*, et un foncteur équipé d'une telle force est alors appelé *foncteur fort*. Cette terminologie est due à Moggi [Mog91], qui considérait des monades fortes pour modéliser les effets dans les langages de programmation. On les trouve également utilisées à des fins plus similaires aux nôtres, c'est-à-dire pour obtenir des relèvements sur des catégories relatives à \mathcal{C}_Γ : voir par

exemple [CF92, CS92, CS95] pour des usages explicites ou [AA99, McC00] pour des usages plus implicites². Ici, la définition d'une force sera la suivante :

Définition 6.2.1 (Force, foncteur fort). *Supposons que \mathcal{C} soit cartésienne, soit $T : \mathcal{C} \rightarrow \mathcal{C}$. Une force pour T est une transformation*

$$\theta_{A,B}^T : A \times T(B) \rightarrow T(A \times B)$$

naturelle en A et B et telle que les diagrammes ci-dessous commutent.

$$\begin{array}{ccccc}
 1 \times T(A) & & (A \times B) \times T(C) & \xrightarrow{\theta_{A \times B, C}^T} & T((A \times B) \times C) \\
 \theta_{1, T(A)}^T \downarrow & \searrow \pi_2 & \alpha_{A, B, T(C)} \downarrow & & \downarrow T(\alpha_{A, B, C}) \\
 T(1 \times A) & \xrightarrow{T(\pi_2)} & T(A) & & T(A \times (B \times C)) \\
 & & A \times (B \times T(C)) & \searrow \text{id}_A \times \theta_{B, C}^T & \nearrow \theta_{A, B \times C}^T \\
 & & & A \times T(B \times C) &
 \end{array}$$

Un foncteur $T : \mathcal{C} \rightarrow \mathcal{C}$ est fort s'il est équipé d'une force θ .

Proposition 6.2.1. *Pour chaque $\Gamma \in \mathcal{C}_0$, $(\theta_{\Gamma, A}^T)_A$ est une loi distributive de $\Gamma \times -$ sur T .*

Démonstration. La compatibilité vis-à-vis de l'effacement de la comonade $\Gamma \times -$ est conséquence du premier diagramme de la définition de force, et de la naturalité de $\theta_{\Gamma, A}^T$ en Γ . Pour montrer la compatibilité vis-à-vis de la duplication, on applique le second diagramme de la définition de force pour $A = B = \Gamma$, et on précompose par $\delta_\Gamma \times T(C)$. On obtient alors le diagramme requis après une commutation entre $\delta_\Gamma \times T(C)$ et $\theta_{\Gamma \times \Gamma, C}^T$, obtenue par naturalité de θ . \square

Ainsi pour chaque Γ , $\theta_{\Gamma, A}^T$ est une loi distributive et en tant que telle engendre un relèvement $T_\Gamma : \mathcal{C}_\Gamma \rightarrow \mathcal{C}_\Gamma$ de $T : \mathcal{C} \rightarrow \mathcal{C}$, qui servira de base pour interpréter la forme étendue de la règle foncteur. On peut se poser la question de la réciproque de la proposition ci-dessus. Si $\theta_{\Gamma, A}^T$ est naturelle en Γ et A et est telle que pour tout $\Gamma \in \mathcal{C}_0$, $(\theta_{\Gamma, A}^T)_A$ est une loi distributive de $\Gamma \times -$ sur T , alors θ acquiert-elle automatiquement le statut de force ? La preuve directe ne semble pas passer et on n'a pas de contre-exemple en vue, laissons donc cette question périphérique en suspens.

Foncteurs fermés. Dans la littérature (notamment [McC96]), on croise parfois une notion de foncteur *fermé* : on en trouve par ailleurs la définition en Section 2.4.1. Pour les besoins de cette discussion, rappelons ici que si \mathcal{C} est une catégorie cartésienne fermée, un foncteur $T : \mathcal{C} \rightarrow \mathcal{C}$ est *fermé* si son action peut être internalisée, c'est-à-dire s'il existe une transformation naturelle

$$\theta_{A,B}^T : A \Rightarrow B \rightarrow T(A) \Rightarrow T(B)$$

telle que pour tout morphisme $f : A \rightarrow B$ on ait ' f' '; $\theta_{A,B}^T = 'T(f)'$. On peut observer que cette notion correspond en fait à l'internalisation dans une catégorie cartésienne fermée de la notion de relèvement de T sur \mathcal{C}_Γ . En effet, les isomorphismes

$$\begin{aligned}
 \mathcal{C}(\Gamma \times A, B) &\cong \mathcal{C}(\Gamma, A \Rightarrow B) \\
 \mathcal{C}(\Gamma \times TA, TB) &\cong \mathcal{C}(\Gamma, TA \Rightarrow TB)
 \end{aligned}$$

²Ces usages plus implicites seront discutés plus précisément en Section 6.2.5.

établissent que passer naturellement de $\mathcal{C}(\Gamma \times A, B)$ à $\mathcal{C}(\Gamma \times TA, TB)$ revient à passer naturellement de $\mathcal{C}(\Gamma, A \Rightarrow B)$ à $\mathcal{C}(\Gamma, TA \Rightarrow TB)$, ce qui se fait en post-composant $\theta_{A,B}^T$ à $f \in \mathcal{C}(\Gamma, A \Rightarrow B)$. Dans une catégorie cartésienne fermée, on peut en fait montrer que cette notion coïncide avec celle de foncteur fort.

6.2.2 Cas contravariant

La situation présentée jusqu'à présent ne s'applique qu'au cas covariant. Or, pour modéliser la règle de foncteur négatif de μLJ , il est nécessaire de savoir également calculer des relèvements pour les endofoncteurs contravariants $N : \mathcal{C}^{op} \rightarrow \mathcal{C}$. Les définitions données ici pour parvenir à ces fins sont nouvelles à la connaissance de l'auteur.

Lois contradistributives. Reprenons la généralité de la Section 1.3.2. Soit donc \mathcal{C} une catégorie, (S, δ, ϵ) une comonade sur \mathcal{C} et $N : \mathcal{C}^{op} \rightarrow \mathcal{C}$ un endofoncteur contravariant. Commençons par adapter la notion de relèvement au cas de tels foncteurs contravariants.

Définition 6.2.2 (Relèvement contravariant). *Un relèvement de $N : \mathcal{C}^{op} \rightarrow \mathcal{C}$ à la catégorie de co-Kleisli \mathcal{C}_S est un foncteur contravariant $N_S : \mathcal{C}_S^{op} \rightarrow \mathcal{C}_S$ tel que pour tout $f : A \rightarrow B$,*

$$N_S(\epsilon_A; f) = SN(f); \epsilon_{NA}$$

La définition qui apparaît naturellement³ quand on examine les relèvements de N sur \mathcal{C}_S est la suivante.

Définition 6.2.3 (Loi contradistributive). *Une loi contradistributive d'une comonade S sur un endofoncteur contravariant $N : \mathcal{C}^{op} \rightarrow \mathcal{C}$ est une transformation naturelle :*

$$\lambda : SNS \xrightarrow{\bullet} N$$

satisfaisant les diagrammes de cohérence suivants :

$$\begin{array}{ccc} SNS & \xrightarrow{\delta_{NSS}} & SSNSS & \xrightarrow{S\lambda S} & SNS & & SN & \xrightarrow{SN\epsilon} & SNS \\ SN\delta \downarrow & & & & \downarrow \lambda & & \epsilon N \downarrow & \swarrow \lambda & \\ SNS & \xrightarrow{\lambda} & N & & N & & N & & N \end{array}$$

Comme pour le cas des foncteurs covariants, les lois contradistributives caractérisent les relèvements des foncteurs contravariants dans la catégorie de co-Kleisli.

Proposition 6.2.2. *Soit (S, δ, ϵ) une comonade sur une catégorie \mathcal{C} et N un endofoncteur contravariant sur \mathcal{C} . On a alors un isomorphisme entre les lois contradistributives $\lambda : SNS \xrightarrow{\bullet} N$ de S sur N et les relèvements de N sur la catégorie de co-Kleisli \mathcal{C}_S .*

Démonstration. Montrons tout d'abord comment une loi contradistributive induit un relèvement. On définit le foncteur $N_S : \mathcal{C}_S^{op}$ de la façon suivante :

- Sur les objets, $N_S(A) = N(A)$;

³Du moins, quand Pierre-Louis Curien est dans les parages : l'auteur le remercie pour avoir esquissé cette définition à partir de celle de contraforce.

– Si $f : SB \rightarrow A$ est un morphisme dans $\mathcal{C}_S^{op}(A, B)$, on définit $N_S(f)$ par :

$$SN(A) \xrightarrow{SN(f)} SNS(B) \xrightarrow{\lambda_B} N(B)$$

Il est immédiat de vérifier que cela définit bien un foncteur. De même, si $N_S : \mathcal{C}_S^{op} \rightarrow \mathcal{C}_S$ est un relèvement de N sur \mathcal{C}_S , on peut définir une famille de morphismes $\lambda_A : SNSA \rightarrow NA$ par :

$$\lambda_A = N_S(id_{SA})$$

Le fait que λ est naturelle et la satisfaction de la cohérence vis-à-vis de la comultiplication de S sont conséquences directes de la functorialité de N_S et de la définition de la composition dans \mathcal{C}_S . La cohérence vis-à-vis de la counité de S correspond exactement à l'équation de la définition de relèvement. Donc λ est bien une loi contradistributive. De plus, ces constructions sont inverses l'une de l'autre : cela revient aux deux équations suivantes, pour tout $f : SB \rightarrow A$.

$$\begin{aligned} SN(id_{SB}); \lambda_B &= \lambda_B \\ SN(f); N_S(id_B) &= N_S(f) \end{aligned}$$

La première est triviale, la seconde utilise la structure de comonade de S , le fait que N_S est un relèvement de N et la définition de la composition dans \mathcal{C}_S . \square

Contraforces. Examinons maintenant le cas particulier qui nous intéresse dans cette thèse, celui où $S = \Gamma \times -$. On obtient alors la notion de *contraforce*, cousine contravariante de la force.

Définition 6.2.4 (Contraforce). *Supposons que \mathcal{C} soit cartésienne, soit $N : \mathcal{C}^{op} \rightarrow \mathcal{C}$. Une contraforce est une famille*

$$\rho_{A,B}^N : A \times N(A \times B) \rightarrow N(B)$$

dinaturelle en A et naturelle en B , et telle que les diagrammes suivants commutent :

$$\begin{array}{ccc} & 1 \times N(A) & \\ & \downarrow 1 \times N(\pi_2) & \searrow \pi_2 \\ & 1 \times N(1 \times A) & \xrightarrow{\rho_{1,A}^N} N(A) \\ & & \\ & B \times (A \times N((A \times B) \times C)) & \\ \nearrow \alpha_{B,A,N((A \times B) \times C)} & & \searrow B \times (A \times N(\alpha_{A,B,C}^{-1})) \\ (B \times A) \times N((A \times B) \times C) & & B \times (A \times N(A \times (B \times C))) \\ \uparrow s_{A,B \times N((A \times B) \times C)} & & \downarrow B \times \rho_{A,B \times C}^N \\ (A \times B) \times N((A \times B) \times C) & & B \times N(B \times C) \\ \searrow \rho_{A \times B,C}^N & & \swarrow \rho_{B,C}^N \\ & N(C) & \end{array}$$

Un foncteur $N : \mathcal{C}^{op} \rightarrow \mathcal{C}$ est fort s'il est équipé d'une contraforce ρ .

Proposition 6.2.3. *Supposons que \mathcal{C} soit cartésienne, soit $N : \mathcal{C}^{op} \rightarrow \mathcal{C}$, et soit $\rho_{\Gamma,A}^N$ une contraforce pour N . Alors pour chaque $\Gamma \in \mathcal{C}_0$, $(\rho_{\Gamma,A}^N)_A$ est une loi distributive de $\Gamma \times -$ sur N .*

Démonstration. C'est évidemment une transformation naturelle en A . La cohérence vis-à-vis de l'affaiblissement $\pi_2 : \Gamma \times A \rightarrow A$ est conséquence de la cohérence de θ vis-à-vis de l'élément neutre 1 pour \times et de la dinaturalité de $\rho_{\Gamma,A}^N$ en Γ . La cohérence vis-à-vis de la duplication vient également de la dinaturalité de ρ en Γ et du second diagramme de cohérence de la définition de contraforce. \square

En conséquence, une contraforce permet bien d'obtenir des relèvements sur \mathcal{C}_Γ d'endofoncteurs contravariants, et ainsi d'interpréter les règles de foncteurs négatifs de μLJ . Comme pour le cas des forces, il ne semble pas évident que toute loi contradistributive de $\Gamma \times -$ sur N qui soit également dinaturelle en Γ soit automatiquement une contraforce.

6.2.3 Relèvement des points fixes

L'objectif des forces n'est pas uniquement de relever des foncteurs, on en a également besoin pour interpréter les règles étendues pour les points fixes, comme rappelées ci-dessous :

$$\frac{\Gamma, P(A) \vdash A}{\Gamma, \mu X.P \vdash A} \mu_l \quad \frac{\Gamma, A \vdash P(A)}{\Gamma, A \vdash \nu X.P} \nu_r$$

Pour ce faire, il faut également relever les structures d'algèbres initiales et de coalgèbres terminales sur \mathcal{C}_Γ . On aura ainsi une interprétation pour ces règles, dont on pourra assurer l'invariance par réduction. Notons que s'il suffisait d'avoir des (contra)forces pour relever les foncteurs dans la section précédente, on verra que ça ne suffit plus pour relever la structure de points fixes. Si on veut pouvoir relever une algèbre initiale d'un endofoncteur T sur \mathcal{C}_S , il va falloir également que S ait un adjoint à droite U , comme c'est d'ailleurs le cas pour $S = \Gamma \times -$.

Algèbres initiales. L'interprétation de la règle μ_l repose sur la proposition suivante :

Proposition 6.2.4. *Soit (S, δ, ϵ) une comonade sur \mathcal{C} , et $T : \mathcal{C} \rightarrow \mathcal{C}$ ayant une algèbre initiale μT , et $\lambda : ST \xrightarrow{\bullet} TS$ une loi distributive de S sur T . Supposons de plus qu'on ait $U : \mathcal{C} \rightarrow \mathcal{C}$ telle⁴ que $S \dashv U$. Alors T_S a également une algèbre initiale dans \mathcal{C}_S .*

Démonstration. Si l'algèbre initiale de T est le couple $(\mu T, \chi_T)$, l'algèbre initiale de T_S est donnée par le couple $(\mu T, \chi'_T)$ où $\chi'_T = \epsilon_{T(\mu T)}; \chi_T$. Notons $\phi_{A,B} : \mathcal{C}(SA, B) \rightarrow \mathcal{C}(A, UB)$ l'isomorphisme de l'adjonction $S \dashv U$. La preuve se déroule comme suit : pour toute algèbre $f : STA \rightarrow A$ de T_S dans \mathcal{C}_S , on construit une algèbre $\hat{f} : T(UA) \rightarrow UA$ de T dans \mathcal{C} , telle que chacun des deux diagrammes suivants commute si et seulement si l'autre commute, pour tout $g : \mu T \rightarrow UA$.

$$\begin{array}{ccc} T(\mu T) & \xrightarrow{T(g)} & T(UA) \\ \chi_T \downarrow & \mathcal{C} & \downarrow \hat{f} \\ \mu T & \xrightarrow{g} & UA \end{array} \quad \begin{array}{ccc} T_S(\mu T) & \xrightarrow{T_S(\phi^{-1}(g))} & T_S(A) \\ \chi'_T \downarrow & \mathcal{C}_S & \downarrow f \\ \mu T & \xrightarrow{\phi^{-1}(g)} & A \end{array}$$

⁴On ne va pas l'utiliser ici, mais notons qu'être adjoint à droite à une comonade donne automatiquement à U une structure de monade.

On définit \hat{f} par $\phi(h)$, où h est obtenu par la composition suivante :

$$STUA \xrightarrow{\delta_{TUA}} SSTUA \xrightarrow{S\lambda_{UA}} STSUA \xrightarrow{ST\epsilon'_A} STA \xrightarrow{f} A$$

où $\epsilon'_A : SUA \rightarrow A$ est la counité de l'adjonction $S \dashv U$. Un calcul direct permet alors de constater que les chemins du haut respectifs de ces deux diagrammes sont reliés par l'isomorphisme ϕ , et de même pour les chemins du bas. Ainsi, la commutation de ces deux diagrammes est équivalente. Mais l'initialité de $(\mu T, \chi_T)$ assure l'unicité du morphisme $g : \mu T \rightarrow UA$ faisant commuter le diagramme de gauche, ainsi on doit également avoir l'unicité de $\phi^{-1}(g)$ faisant commuter le diagramme de droite. \square

En particulier, si \mathcal{C} est cartésienne fermée et si $T : \mathcal{C} \rightarrow \mathcal{C}$ est un foncteur fort, alors les conditions de cette proposition sont satisfaites et pour tout $\Gamma \in \mathcal{C}_0$, on sait que si T avait une algèbre initiale dans \mathcal{C} , son relèvement T_Γ à \mathcal{C}_Γ en aura également une.

Coalgèbres terminales. La situation des coalgèbres terminales est légèrement plus simple : inutile de supposer que S a un adjoint à droite. La proposition suivante permettra de donner une interprétation de la règle ν_r :

Proposition 6.2.5. *Soit (S, δ, ϵ) une comonade sur \mathcal{C} , et $T : \mathcal{C} \rightarrow \mathcal{C}$ ayant une coalgèbre terminale νT , et $\lambda : ST \rightarrow TS$ une loi distributive de S sur T . Alors T_S a également une coalgèbre terminale dans \mathcal{C}_S .*

Démonstration. Si $(\nu T, \chi_T)$ est la coalgèbre terminale de T dans \mathcal{C} on construit la coalgèbre terminale de T_S dans \mathcal{C}_S par le couple $(\nu T, \chi'_T)$ où $\chi'_T = \epsilon_{\nu T}; \chi_T$. Pour toute coalgèbre $f : SA \rightarrow TA$ de T_S dans \mathcal{C}_S , on construit $\hat{f} : SA \rightarrow T(SA)$ par :

$$SA \xrightarrow{\delta_A} SSA \xrightarrow{Sf} STA \xrightarrow{\lambda_A} TSA$$

Alors, la définition du relèvement de T permet de montrer que chacun des deux diagrammes suivants commute si et seulement si l'autre commute, pour tout $g : SA \rightarrow \nu T$:

$$\begin{array}{ccc} SA & \xrightarrow{g} & \nu T \\ \downarrow \hat{f} & \searrow c & \downarrow \chi_T \\ T(SA) & \xrightarrow{T(g)} & T(\nu T) \end{array} \quad \begin{array}{ccc} A & \xrightarrow{g} & \nu T \\ \downarrow f & \searrow c_S & \downarrow \chi'_T \\ T_S(A) & \xrightarrow{T_S(g)} & T_S(\nu T) \end{array}$$

d'où on déduit immédiatement que $(\nu T, \chi'_T)$ est bien une coalgèbre terminale pour T_S . \square

Comme dans le paragraphe précédent, le cas qui va le plus nous intéresser est celui où \mathcal{C} est cartésienne fermée, $T : \mathcal{C} \rightarrow \mathcal{C}$ est un foncteur fort et pourvu d'une coalgèbre terminale $(\nu T, \chi_T)$. Dans ce cas, la proposition ci-dessus s'applique et pour tout $\Gamma \in \mathcal{C}_0$, la coalgèbre terminale de T se relève en une coalgèbre terminale de T_Γ dans \mathcal{C}_Γ .

6.2.4 Uniformité du relèvement

Soit $\Gamma \in \mathcal{C}_0$. Les sections précédentes nous ont appris comment relever les endofoncteurs forts de \mathcal{C} sur \mathcal{C}_Γ , avec leurs éventuelles algèbres initiales et coalgèbres terminales. Malheureusement, cette structure n'est pas encore assez contrainte pour fournir un modèle de μLJ .

Expansion de foncteurs. Le problème vient des règles d'expansion de foncteurs. Prenons comme exemple la règle (**Fonc.3a**). Si P_1 et P_2 sont deux types positifs, toute application de la règle $[P_1 \times P_2]$ peut s'expanser en :

$$\frac{\frac{\frac{\pi}{\Gamma, A \vdash B}}{\Gamma, P_1(A) \vdash P_1(B)} [P_1]}{\Gamma, P_1(A), P_2(A) \vdash P_1(B)} S \quad \frac{\frac{\frac{\pi}{\Gamma, A \vdash B}}{\Gamma, P_2(A) \vdash P_2(B)} [P_2]}{\Gamma, P_1(A), P_2(A) \vdash P_2(B)} S}{\frac{\Gamma, P_1(A) \times P_2(A) \vdash P_1(B)}{\Gamma, P_1(A) \times P_2(A) \vdash P_2(B)} \times_l \quad \frac{\Gamma, P_1(A), P_2(A) \vdash P_1(B)}{\Gamma, P_1(A) \times P_2(A) \vdash P_2(B)} \times_r} S$$

À première vue, cette expansion semble correspondre avec le fait que dans toute catégorie cartésienne fermée, si $P_1, P_2 : \mathcal{C} \rightarrow \mathcal{C}$, on a par définition de l'action bifonctorielle de \times pour tout $f : A \rightarrow B$:

$$(P_1 \times P_2)(f) = \langle \pi_1; P_1(f), \pi_2; P_2(f) \rangle$$

Il serait donc trivial que deux preuves convertibles par cette règle de réduction s'interprètent par le même morphisme. Cette observation est parfaitement valide avec les règles affaiblies de μLJ , mais ici il faut que les règles d'expansion soient valides pour les règles relatives à un contexte Γ non vide, autrement dit il faut que ces équations d'expansion soient valides non seulement pour les foncteurs, mais pour leurs relèvements sur \mathcal{C}_Γ . Pour valider la règle d'expansion de $[P_1 \times P_2]$, il faudra donc au préalable valider l'égalité suivante :

$$(S \times T)_\Gamma = S_\Gamma \times T_\Gamma$$

De par la définition des relèvements sur \mathcal{C}_Γ , une telle équation se ramène à l'équation suivante sur les forces de S, T et $S \times T$:

$$\theta_{\Gamma, A}^{S \times T} = \langle \Gamma \times \pi_1; \theta_{\Gamma, A}^S, \Gamma \times \pi_2; \theta_{\Gamma, A}^T \rangle$$

Une façon évidente d'obtenir cette équation est de la poser comme définition : si on définit les forces par induction sur les types, on sera gratuitement assurés de ce genre de propriétés d'uniformité. Cependant, on verra en Section 6.2.5 que la gestion des types (co)inductifs entrelacés rend cette option (pour l'instant) impraticable. Par exemple, il est très difficile dans notre contexte de construire une force pour une algèbre initiale paramétrée T^μ à partir d'une force du foncteur T . On a donc opté dans cette thèse pour une option alternative : on suppose dans le modèle catégorique qu'on a les conditions d'uniformité voulues, et on vérifie ultérieurement que ces conditions sont bien validées dans le modèle (concret) de jeux.

Types forts. La terminologie de *type fort* est inspirée de [CS92, CS95]. L'idée est que chaque type (foncteur) vient automatiquement équipé d'une force, et ces forces vérifient un certain nombre de propriétés d'uniformité :

Définition 6.2.5 (Types forts). *Une catégorie \mathcal{C} avec des types forts est une catégorie cartésienne fermée avec des coproduits faibles fonctoriels et équipée d'une classe \mathcal{F} de foncteurs :*

$$T : (\mathcal{C}^{op})^k \times \mathcal{C}^n \rightarrow \mathcal{C}$$

vérifiant les propriétés suivantes :

- \mathcal{F} contient l'identité, les foncteurs constants ainsi que les constructeurs de base $-_1 + -_2$, $-_1 \times -_2$ et $-_1 \Rightarrow -_2$;
- \mathcal{F} est stable par composition (au sens des opérades [Lei04]) ;
- \mathcal{F} est stable par contraction : si $F(-_1, -_2, -_3) : \mathcal{C} \times \mathcal{C} \times \mathcal{D} \rightarrow \mathcal{C}$ est dans \mathcal{F} , alors $F(-_1, -_1, -_3) : \mathcal{C} \times \mathcal{D} \rightarrow \mathcal{C}$. De même avec \mathcal{C}^{op} à la place de \mathcal{C} à gauche.

Ceux de ces foncteurs qui sont unaires (i.e. $P : \mathcal{C} \rightarrow \mathcal{C}$ ou $N : \mathcal{C}^{op} \rightarrow \mathcal{C}$) sont forts, c'est-à-dire que, suivant leur variance, ils ont une force ou une contraforce. Ces données doivent vérifier les propriétés additionnelles suivantes. Dans cette définition, on désigne par P les foncteurs covariants et N les foncteurs contravariants.

- Les familles $\theta_{\Gamma,A}^T$ et $\rho_{\Gamma,A}^T$ sont également naturelles en T ;
- Compatibilité avec l'identité et les constantes :

$$\begin{aligned}\theta_{\Gamma,A}^{Id} &= id_{\Gamma \times A} \\ \theta_{\Gamma,A}^B &= \pi_2 \\ \rho_{\Gamma,A}^B &= \pi_2\end{aligned}$$

- Compatibilité avec la composition et la contraction :

$$\begin{aligned}\theta_{\Gamma,A}^{P_1 P_2} &= \theta_{\Gamma, P_2(A)}^{P_1}; P_1(\theta_{\Gamma,A}^{P_2}) \\ \rho_{\Gamma,A}^{PN} &= \theta_{\Gamma, N(\Gamma \times A)}^P; P(\rho_{\Gamma,A}^N) \\ \rho_{\Gamma,A}^{NP} &= \Gamma \times N(\theta_{\Gamma,A}^P); \rho_{\Gamma, P(A)}^N \\ \theta_{\Gamma,A}^{N_1 N_2} &= \Gamma \times N_1(\rho_{\Gamma,A}^{N_2}); \rho_{\Gamma, N_2(\Gamma \times A)}^{N_1} \\ \theta_{\Gamma,A}^{P(-,-)} &= \langle \pi_1, \theta_{\Gamma,A}^{P(-,A)} \rangle; \theta_{\Gamma,A}^{P(\Gamma \times A, -)} \\ \rho_{\Gamma,A}^{N(-,-)} &= \langle \pi_1, \rho_{\Gamma,A}^{N(-, \Gamma \times A)} \rangle; \rho_{\Gamma,A}^{N(A, -)}\end{aligned}$$

- Compatibilité avec les sommes et la structure fermée.

$$\begin{aligned}\theta_{\Gamma,A}^{-+-} &= \gamma_{\Gamma,A,A} \\ \rho_{\Gamma,A}^{-\Rightarrow C} &= \Lambda(\langle \langle \pi_2; \pi_1, \pi_1 \rangle, \pi_2; \pi_2 \rangle; ev) \\ \theta_{\Gamma,A}^{C \Rightarrow -} &= \Lambda(\langle \pi_2; \pi_1, \langle \pi_1, \pi_2; \pi_2 \rangle; ev \rangle)\end{aligned}$$

On rappelle que $\gamma_{\Gamma,A,A}$ est la loi de distributivité du produit sur la somme construite dans les préliminaires. On dit que \mathcal{C} a les types pseudo-polynomiaux forts si \mathcal{F} contient $- \Rightarrow A$ pour toute formule A mais pas nécessairement $-_1 \Rightarrow -_2$, et si on oublie la dernière équation.

Types pseudo-polynomiaux forts. De par les limitations du modèle de jeux, on parlera occasionnellement le long de ce chapitre des types *pseudo-polynomiaux*. Ils correspondent à la restriction du modèle catégorique au fragment *pseudo-polynomial* de μLJ , défini en Définition 5.1.7.

Remarque. Il est possible de se passer de l'équation reliant la force à la loi de distributivité du produit sur la somme, mais cela implique de redéfinir le relèvement des coproduits faibles fonctoriels à \mathcal{C}_Γ en utilisant uniquement la structure de types forts, sans utiliser $\gamma_{\Gamma,A,B}$ comme en Section 1.2.2. S'ensuivent alors des preuves sans grand intérêt mais d'une grande technicité qu'on préfère éviter ici.

Expansion de foncteurs. Commençons par montrer que la structure de types forts implique toutes les équations naturelles entre forces.

Lemme 6.2.6. *Soit \mathcal{C} une catégorie à types forts (éventuellement pseudo-polynomiaux). On a alors les décompositions des forces suivantes :*

$$\begin{aligned}\theta_{\Gamma,A}^{P \times P'} &= \langle \Gamma \times \pi_1; \theta_{\Gamma,A}^P, \Gamma \times \pi_2; \theta_{\Gamma,A}^{P'} \rangle \\ \theta_{\Gamma,A}^{P+P'} &= \gamma_{\Gamma, P(A), P'(A)}; \theta_{\Gamma,A}^P + \theta_{\Gamma,A}^{P'} \\ \theta_{\Gamma,A}^{N \Rightarrow P} &= \Lambda(\langle \pi_2; \pi_1, \langle \langle \pi_2; \pi_1, \pi_1 \rangle; \rho_{\Gamma,A}^N, \pi_2; \pi_2 \rangle; ev \rangle; \theta_{\Gamma,A}^P) \\ \rho_{\Gamma,A}^{P \Rightarrow N} &= \Lambda(\langle \pi_2; \pi_1, \langle \langle \pi_2; \pi_1, \pi_1 \rangle; \theta_{\Gamma,A}^P, \pi_2; \pi_2 \rangle; ev \rangle; \rho_{\Gamma,A}^N)\end{aligned}$$

Ainsi que les équations identiques pour $\rho_{\Gamma,A}^{N \times N'}$ et $\rho_{\Gamma,A}^{N+N'}$.

Démonstration. Pour le produit, il suffit d'exploiter la naturalité de $\theta_{\Gamma,A}^{P \times P'}$ en $P \times P'$ et la naturalité des projections π_1 et π_2 . Pour la somme, on utilise la compatibilité des forces avec composition et la loi de distributivité. Pour la flèche, il s'agit de calculs directs utilisant la compatibilité des forces avec la contraction, la composition et la structure cartésienne fermée. \square

Proposition 6.2.7. *Soit \mathcal{C} une catégorie à types forts (éventuellement pseudo-polynomiaux). On a les équations suivantes :*

$$\begin{aligned}(P \times P')_\Gamma &= P_\Gamma \times P'_\Gamma \\ (N \times N')_\Gamma &= N_\Gamma \times N'_\Gamma \\ (P + P')_\Gamma &= P_\Gamma + P'_\Gamma \\ (N + N')_\Gamma &= N_\Gamma + N'_\Gamma \\ (N \Rightarrow P)_\Gamma &= N_\Gamma \Rightarrow P_\Gamma \\ (P \Rightarrow N)_\Gamma &= P_\Gamma \Rightarrow N_\Gamma\end{aligned}$$

Démonstration. Ces équations s'obtiennent directement à partir du lemme précédent. \square

Relèvements d'arité supérieure. À priori, les forces et contraforces ne permettent de relever naturellement sur \mathcal{C}_Γ que les foncteurs unaires. Terminons cette sous-section par la preuve qu'en fait, ces relèvements s'étendent de façon canonique aux foncteurs d'arité supérieure à 1, et ce même s'ils sont de variance mixte. Ce fait repose sur le lemme suivant :

Lemme 6.2.8 (Commutations). *Soit \mathcal{C} une catégorie avec les types forts (éventuellement pseudo-polynomiaux), soit $T \in \mathcal{F}$. On a les règles de commutation suivantes :*

(1) Si $T : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$:

$$\langle \pi_1, \theta_{\Gamma,B}^{T(A,-)} \rangle; \theta_{\Gamma,A}^{T(-,\Gamma \times B)} = \langle \pi_1, \theta_{\Gamma,A}^{T(-,B)} \rangle; \theta_{\Gamma,B}^{T(\Gamma \times A,-)}$$

(2) Si $T : \mathcal{C}^{op} \times \mathcal{C} \rightarrow \mathcal{C}$:

$$\langle \pi_1, \theta_{\Gamma,B}^{T(\Gamma \times A,-)} \rangle; \rho_{\Gamma,A}^{T(-,\Gamma \times B)} = \langle \pi_1, \rho_{\Gamma,A}^{T(-,B)} \rangle; \theta_{\Gamma,B}^{T(A,-)}$$

(3) Si $T : \mathcal{C}^{op} \times \mathcal{C}^{op} \rightarrow \mathcal{C}$:

$$\langle \pi_1, \rho_{\Gamma, B}^{T(\Gamma \times A, -)} \rangle; \rho_{\Gamma, A}^{T(-, B)} = \langle \pi_1, \rho_{\Gamma, A}^{T(-, \Gamma \times B)} \rangle; \rho_{\Gamma, B}^{T(A, -)}$$

Démonstration. Détaillons la preuve de (1), les autres sont similaires. On commence par remarquer que par functorialité de T , la famille $(T(A, f))_A$ est naturelle en A . Par naturalité de θ en T , on en déduit que le diagramme suivant commute, pour tout f :

$$\begin{array}{ccc} \Gamma \times T(A, B) & \xrightarrow{\Gamma \times T(A, f)} & \Gamma \times T(A, C) \\ \theta_{\Gamma, A}^{T(-, B)} \downarrow & & \downarrow \theta_{\Gamma, A}^{T(-, C)} \\ T(\Gamma \times A, B) & \xrightarrow{T(\Gamma \times A, f)} & T(\Gamma \times A, C) \end{array}$$

Mais ce diagramme exprime exactement le fait que la famille suivante est une transformation naturelle :

$$\left(\theta_{\Gamma, A}^{T(-, B)} \right)_B : \Gamma \times T(A, -) \xrightarrow{\bullet} T(\Gamma \times A, -)$$

Par conséquent, une autre utilisation de la naturalité de θ en T donne l'égalité suivante :

$$\theta_{\Gamma, B}^{\Gamma \times T(A, -)}; \theta_{\Gamma, A}^{T(-, \Gamma \times B)} = \Gamma \times \theta_{\Gamma, A}^{T(-, B)}; \theta_{\Gamma, B}^{T(\Gamma \times A, -)}$$

On en déduit immédiatement l'équation voulue, grâce à la décomposition du Lemme 6.2.6. \square

Proposition 6.2.9. *Considérons un foncteur $T : (\mathcal{C}^{op})^k \times \mathcal{C}^n \rightarrow \mathcal{C}$ dans \mathcal{F} . Alors, il existe un unique foncteur $T_\Gamma : (\mathcal{C}_\Gamma^{op})^k \times \mathcal{C}_\Gamma^n \rightarrow \mathcal{C}_\Gamma$ tel que pour tout $i \in \{1, \dots, p, p+1, \dots, p+k\}$, pour tous A_1, \dots, A_{k+p-1} dans \mathcal{C} ,*

$$(T(A_1, A_2, \dots, -i, \dots, A_{k+n}))_\Gamma = T_\Gamma(A_1, A_2, \dots, -i, \dots, A_{k+n})$$

Démonstration. La définition se fait composante par composante. Les commutations nécessaires sont conséquences immédiates du Lemme 6.2.8. \square

6.2.5 Discussion : construction des forces

À ce point, on espère avoir convaincu le lecteur de deux points fondamentaux :

1. Il est nécessaire pour interpréter μLJ que tous les foncteurs considérés soient *forts* ;
2. Les règles d'expansion imposent d'avoir une *uniformité* sur ces forces.

Pour satisfaire ces deux conditions, il est suffisant d'interpréter le calcul dans une catégorie à types forts. Il est cependant raisonnable d'objecter que cette structure semble lourde, et largement redondante : les contraintes d'uniformité données semblent en fait suivre les contours d'une simple définition des forces par induction sur les types. Pour justifier cette structure, commençons donc par la remettre en question et nous poser la question suivante : «où construire les forces?»

$$\frac{\frac{\theta^T(A)_{\Gamma, \Gamma \Rightarrow Z}}{\Gamma, T(A)[\Gamma \Rightarrow Z/Y] \vdash T(A)[\Gamma \times (\Gamma \Rightarrow Z)/Y]} \quad \frac{\frac{ev}{\Gamma \times (\Gamma \Rightarrow Z) \vdash Z}}{T(A)[\Gamma \times (\Gamma \Rightarrow Z/Y) \vdash T(A)[Z/Y]]} [T(A)[Y]]}{\Gamma, T(A)[\Gamma \Rightarrow \mu Y.T(\Gamma \times A)/Y] \vdash \mu Y.T(\Gamma \times A)} Cut$$

$$\frac{\frac{\Gamma, T(A)[\Gamma \Rightarrow \mu Y.T(\Gamma \times A)/Y] \vdash \mu Y.T(\Gamma \times A)}{T(A)[\Gamma \Rightarrow \mu Y.T(\Gamma \times A)/Y] \vdash \Gamma \Rightarrow \mu Y.T(\Gamma \times A)} \Rightarrow_r}{\frac{\mu Y.T(A) \vdash \Gamma \Rightarrow \mu Y.T(\Gamma \times A)}{\Gamma, \mu Y.T(A) \vdash \mu Y.T(\Gamma \times A)} \Rightarrow_e} \mu_i$$

FIG. 6.1 – Candidat de force pour un foncteur positif défini par point fixe.

Dans la syntaxe. La solution la plus directe pour avoir accès aux règles étendues de μLJ , serait de les dériver directement dans la syntaxe à partir des règles affaiblies. Autrement dit, il s'agirait de construire explicitement des forces comme des opérateurs syntaxiques, et de s'en servir pour définir les règles étendues : notons que c'est plus ou moins l'approche de [AA99], et c'était l'approche défendue (mais pas développée) dans [Cla09a]. Il est effectivement parfaitement possible, à partir des règles affaiblies de μLJ , de définir par induction sur les types des candidats pour les forces. Le problème est alors de vérifier que ces forces sont sémantiquement correctes, c'est à dire qu'elles engendrent bien la bonne dynamique. C'est facile dans le contexte syntaxique (avec liaison et substitution implicite) de [AA99], mais ça devient un enfer dans μLJ où on se heurte à l'extrême complexité de la définition de ces forces, notamment pour les foncteurs définis par point fixe. Pour en donner une idée, donnons en Figure 6.1 un aperçu de la dérivation d'un tel candidat de force. Il est très probablement vrai que ces candidats sont bel et bien des forces et qu'elles forment une structure uniforme, mais c'est difficile à montrer.

Dans le modèle catégorique. Ce serait sûrement la solution la plus satisfaisante puisque la plus générale. On a pas encore pu démontrer la conjecture suivante, mais les travaux présentés sur les types forts permettent au moins de la formuler.

Conjecture 6.2.10. *Toute catégorie faiblement μ -fermée est μ -fermée, autrement dit on peut dériver automatiquement une structure de types forts comprenant les algèbres initiales paramétrées et coalgèbres terminales paramétrées.*

Dans la sémantique de jeux. Dans le cas qui nous occupe, c'est l'approche la plus facile. En effet, l'interprétation des règles foncteurs donnera lieu dans la sémantique de jeux à la construction d'une classe \mathcal{F} de foncteurs qu'on appellera *foncteurs ouverts*, qui équipera sans mal la catégorie **Inn** des arènes et stratégies innocentes d'une structure de types forts. Notons que dans le modèle de jeux des types récursifs de McCusker [McC00], c'est également directement dans la catégorie de jeux que sont construites les forces (plus précisément la structure «fermée»). Par exemple, les forces des invariants minimaux paramétrés y sont construites par une limite d'une suite croissante de stratégies. Aucune uniformité sur les forces n'y est demandée, mais par ailleurs le modèle n'a pas à valider les expansions de foncteurs qui n'entrent pas en jeu dans la réduction de FPC.

6.3 Modélisation de μLJ

Dans cette dernière section, achevons la construction du modèle de μLJ . On va commencer par donner la définition d'une catégorie μ -fermée, ainsi que quelques lemmes s'y rapportant. Puis on y donnera l'interprétation de μLJ .

6.3.1 Catégories μ -fermées

Dans la définition qui suit, on isole les cas où on ne s'intéresse qu'aux formules *strictes* ou *pseudo-polynomiales* : le modèle de jeux sera restreint à ces classes de formules.

Définition 6.3.1 (Catégorie μ -fermée). *Une catégorie \mathcal{C} est μ -fermée (pour les formules strictes, pseudo-polynomiales) si :*

- \mathcal{C} a les types (pseudo-polynomiaux) forts ;
- La fonction d'interprétation $\llbracket - \rrbracket$ est totale sur \mathcal{F} (pour les formules strictes, pseudo-polynomiales). Si \mathcal{C} est μ -fermée, notons \mathfrak{F} l'image de $\llbracket - \rrbracket$ dans \mathcal{F} .

Remarque. Explicitons le sens des parenthèses dans cette définition. On fixe la convention que chacune des conditions entre parenthèses est vérifiée dans l'en-tête de l'énoncé si chacune de ses occurrences est vérifiée dans le corps de l'énoncé. Par exemple, on a en fait défini ici quatre notions :

- Les catégories μ -fermées, qui ont les types forts et une interprétation $\llbracket - \rrbracket$ totale sur toutes les formules de μLJ ;
- Les catégories μ -fermées pour les formules pseudo-polynomiales, qui ont les types pseudo-polynomiaux forts et une interprétation $\llbracket - \rrbracket$ totale sur les formules du fragment pseudo-polynomial de μLJ ;
- Les catégories μ -fermées pour les formules strictes, qui ont les types forts et une interprétation $\llbracket - \rrbracket$ totale sur les formules du fragment strict de μLJ ;
- Les catégories μ -fermées pour les formules strictes et pseudo-polynomiales, qui ont les types pseudo-polynomiaux forts et une interprétation $\llbracket - \rrbracket$ totale sur les formules du fragment strict et pseudo-polynomial de μLJ .

On utilisera cette convention de notation pour certains énoncés à venir, sans forcément l'explicitier comme ici.

Une catégorie \mathcal{C} est μ -fermée si elle est faiblement μ -fermée et si l'image de sa fonction d'interprétation $\llbracket - \rrbracket$ est incluse dans l'ensemble \mathcal{F} des foncteurs forts de \mathcal{C} . Pour montrer que les catégories μ -fermées sont un modèle de μLJ , il faut montrer qu'elles valident également les équations correspondant à l'expansion des algèbres initiales paramétrées et coalgèbres terminales paramétrées, c'est-à-dire qu'il faut montrer des équations du genre :

$$(T_{\Gamma})^{\mu} = (T^{\mu})_{\Gamma}$$

Avant de démontrer cette égalité, expliquons pourquoi le terme de gauche a un sens : justifions que T_{Γ} a nécessairement toujours une algèbre initiale paramétrée ou une coalgèbre terminale paramétrée si T en avait une. On passera ensuite aux expansions proprement dites.

Proposition 6.3.1. *Soit \mathcal{D} une catégorie de la forme $(\mathcal{C}^{op})^k \times \mathcal{C}^n$. Dans ce cas, on va noter sans ambiguïté $\mathcal{D}_{\Gamma} = (\mathcal{C}_{\Gamma}^{op})^k \times (\mathcal{C}_{\Gamma})^n$. Soit $T : \mathcal{C} \times \mathcal{D} \rightarrow \mathcal{C}$ dans \mathcal{F} . Si T a une algèbre initiale paramétrée $T^{\mu} : \mathcal{D} \rightarrow \mathcal{C}$ dans \mathcal{C} , alors T_{Γ} a une algèbre initiale paramétrée $(T_{\Gamma})^{\mu} : \mathcal{D}_{\Gamma} \rightarrow \mathcal{C}_{\Gamma}$. De*

même, si T a une coalgèbre terminale paramétrée T^ν , T_Γ a une coalgèbre terminale paramétrée $(T_\Gamma)^\nu$.

Démonstration. C'est une conséquence directe des Propositions 6.2.4 et 6.2.5, ainsi que de la Proposition 1.3.2 définissant les algèbres initiales et coalgèbres terminales paramétrées. \square

Corollaire 6.3.2. Soit \mathcal{D} une catégorie de la forme $(\mathcal{C}^{op})^k \times \mathcal{C}^n$, on notera $\mathcal{D}_\Gamma = (\mathcal{C}_\Gamma^{op})^k \times \mathcal{C}_\Gamma^n$. Soit $T : \mathcal{C} \times \mathcal{D} \rightarrow \mathcal{C}$ dans \mathfrak{F} . Alors, on a les équations suivantes :

$$\begin{aligned} (T^\mu)_\Gamma &= (T_\Gamma)^\mu \\ (T^\nu)_\Gamma &= (T_\Gamma)^\nu \end{aligned}$$

Démonstration. Traitons la première égalité, pour le cas où $T : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$. Les autres cas sont similaires ou s'en déduisent. On rappelle que pour tout $f : \Gamma \times A \rightarrow B$, $(T_\Gamma)^\mu(f)$ est défini (par la Proposition 1.3.2) comme l'unique morphisme h tel que :

$$\begin{array}{ccc} T(T^\mu(A), A) & \xrightarrow{T_\Gamma(-, A)(h)} & T(T^\mu(B), A) \\ \downarrow \chi_{T_\Gamma(-, A)}^\Gamma & \text{c}_\Gamma & \downarrow (T_\Gamma(T_\Gamma^\mu(B), -))(f) \\ T^\mu(A) & \xrightarrow{h} & T^\mu(B) \\ & & \downarrow i_{T_\Gamma(-, B)}^\Gamma \end{array}$$

commute. Pour montrer que $(T^\mu)_\Gamma(f) = (T_\Gamma)^\mu$, il suffit donc de montrer que $h = (T^\mu)_\Gamma(f)$ fait commuter ce diagramme, ce qui est assuré par un calcul fastidieux utilisant la naturalité de θ en T pour pouvoir commuter θ et χ (χ étant naturelle par construction) et la compatibilité de θ avec la contraction et la composition. \square

6.3.2 Lemmes de substitution

Avant de passer à l'interprétation proprement dite, il nous faut montrer quelques lemmes assurant l'invariance de l'interprétation par l'opération de substitution de μLJ .

Lemme 6.3.3 (Substitution sous μ). Soit $f : \Gamma \times T(A) \rightarrow A$ et $\delta : \Delta \rightarrow \Gamma$. Notons $(f)_\Gamma^\dagger$ le morphisme obtenu par l'action de l'algèbre initiale de T_Γ sur \mathcal{C}_Γ . Alors :

$$\delta \times \mu T; (f)_\Gamma^\dagger = (\delta \times T(A); f)_\Delta^\dagger$$

Démonstration. Rappelons que $(\delta \times T(A); f)_\Delta^\dagger$ est par définition l'unique morphisme d'algèbre entre l'algèbre initiale $(\mu T, \chi_T^\Delta)$ et l'algèbre $(A, \delta \times T(A); f)$ sur \mathcal{C}_Δ . Pour montrer l'équation ci-dessus, il suffit donc de montrer que $\delta \times \mu T; (f)_\Gamma^\dagger$ est un tel morphisme d'algèbre, c'est-à-dire que le carré ci-dessous commute dans \mathcal{C}_Δ :

$$\begin{array}{ccc}
T(\mu T) & \xrightarrow{T_{\Delta}(\delta \times \mu T; (f)_{\Gamma}^{\dagger})} & T(A) \\
\chi_T^{\Delta} \downarrow & & \downarrow \delta \times T(A); f \\
\mu T & \xrightarrow{\delta \times \mu T; (f)_{\Gamma}^{\dagger}} & A
\end{array}$$

Cette égalité se montre par un calcul direct. \square

Lemme 6.3.4 (Substitution sous ν). *Soit $f : \Gamma \times A \rightarrow T(A)$ et $\delta : \Delta \rightarrow \Gamma$. Notons $(f)_{\Gamma}^{\dagger}$ le morphisme obtenu par l'action de la coalgèbre terminale de T_{Γ} sur \mathcal{C}_{Γ} . Alors :*

$$\delta \times A; (f)_{\Gamma}^{\dagger} = (\delta \times A; f)_{\Delta}^{\dagger}$$

Démonstration. Même raisonnement que pour le lemme précédent, il suffit de montrer que le carré suivant commute dans \mathcal{C}_{Δ} .

$$\begin{array}{ccc}
A & \xrightarrow{\delta \times A; (f)_{\Gamma}^{\dagger}} & \nu T \\
\delta \times A; f \downarrow & & \downarrow (\chi_T^{\Delta})^{-1} \\
T(A) & \xrightarrow{T_{\Delta}(\delta \times A; (f)_{\Gamma}^{\dagger})} & T(\nu T)
\end{array}$$

qui se montre par calcul direct. \square

Lemme 6.3.5 (Substitution sous les foncteurs). *Soient $T : \mathcal{C} \rightarrow \mathcal{C}$ et $N : \mathcal{C}^{op} \rightarrow \mathcal{C}$ des foncteurs de \mathcal{F} , et soient $\delta : \Delta \rightarrow \Gamma$ et $f : \Gamma \times A \rightarrow B$. Alors :*

$$\begin{aligned}
\delta \times T(A); T_{\Gamma}(f) &= T_{\Delta}(\delta \times A; f) \\
\delta \times T(B); N_{\Gamma}(f) &= N_{\Delta}(\delta \times A; f)
\end{aligned}$$

Démonstration. Immédiat par naturalité et dinaturalité des forces et contraforces et par définition de T_{Γ} et N_{Γ} . \square

6.4 Interprétation

6.4.1 Interprétation d'une preuve

Comme d'habitude, une preuve d'un séquent $\Gamma_1, \dots, \Gamma_n \vdash A$ sera interprétée comme un morphisme $[[\Gamma_1]] \times \dots \times [[\Gamma_n]] \rightarrow [[A]]$ (par défaut fixons que ces produits successifs sont parenthésés à gauche). Notons que certains morphismes présents dans cette interprétation servent seulement à la gestion du contexte. Lorsqu'on peut obtenir Δ de Γ par simple réordonnement du contexte, mais que l'isomorphisme correspondant ne peut être obtenu de façon concise par la structure cartésienne, on le notera $\gamma_{\Delta}^{\Gamma} : \Gamma \rightarrow \Delta$, ou simplement γ si ses domaines et codomains sont clairs. Notons que sa définition formelle est directe à partir de Γ et Δ . Soit π une preuve d'un séquent $\Gamma \vdash A$, on définit $[[\pi]] : [[\Gamma]] \rightarrow [[A]]$ par induction, comme suit.

– Axiome.

$$\llbracket \frac{}{A \vdash A} ax \rrbracket = id_A$$

– Coupure.

$$\begin{aligned} & \left[\frac{\frac{\pi_1}{\Gamma \vdash A} \quad \frac{\pi_2}{\Delta, A \vdash B}}{\Gamma, \Delta \vdash B} Cut \right] \\ = & \llbracket \Gamma, \Delta \rrbracket \xrightarrow{\gamma} \llbracket \Delta \rrbracket \times \llbracket \Gamma \rrbracket \xrightarrow{\llbracket \Delta \rrbracket \times \llbracket \pi_1 \rrbracket} \llbracket \Delta \rrbracket \times \llbracket A \rrbracket \xrightarrow{\llbracket \pi_2 \rrbracket} \llbracket B \rrbracket \end{aligned}$$

– Contraction.

$$\left[\frac{\frac{\pi}{\Gamma, A, A \vdash B} C}{\Gamma, A \vdash B} \right] = \llbracket \Gamma, A \rrbracket \xrightarrow{\langle id, \pi_2 \rangle} \llbracket \Gamma, A, A \rrbracket \xrightarrow{\llbracket \pi \rrbracket} \llbracket B \rrbracket$$

– Affaiblissement.

$$\left[\frac{\frac{\pi}{\Gamma \vdash B} W}{\Gamma, A \vdash B} \right] = \llbracket \Gamma, A \rrbracket \xrightarrow{\pi_1} \llbracket \Gamma \rrbracket \xrightarrow{\llbracket \pi \rrbracket} \llbracket B \rrbracket$$

– Échange.

$$\left[\frac{\frac{\pi}{\Gamma, A, B, \Delta \vdash C} \gamma}{\Gamma, B, A, \Delta \vdash C} \right] = \llbracket \Gamma, A, B, \Delta \rrbracket \xrightarrow{\gamma} \llbracket \Gamma, B, A, \Delta \rrbracket \xrightarrow{\llbracket \pi \rrbracket} \llbracket C \rrbracket$$

– Introduction de l'implication à droite.

$$\left[\frac{\frac{\pi}{\Gamma, A \vdash B} \Rightarrow_r}{\Gamma \vdash A \Rightarrow B} \right] = \llbracket \Gamma \rrbracket \xrightarrow{\Lambda(\langle \pi_2, \pi_1 \rangle; \llbracket \pi \rrbracket)} \llbracket A \Rightarrow B \rrbracket$$

– Introduction de l'implication à gauche.

$$\begin{aligned} & \left[\frac{\frac{\frac{\pi_1}{\Gamma \vdash A} \quad \frac{\pi_2}{\Delta, B \vdash C}}{\Gamma, \Delta, A \Rightarrow B \vdash C} \Rightarrow_l}{\Gamma, \Delta, A \Rightarrow B} \right] = \\ & \llbracket \Gamma, \Delta, A \Rightarrow B \rrbracket \xrightarrow{\gamma} \llbracket \Delta \rrbracket \times (\llbracket \Gamma \rrbracket \times (\llbracket A \rrbracket \Rightarrow \llbracket B \rrbracket)) \\ & \xrightarrow{\llbracket \Delta \rrbracket \times (\llbracket \pi_1 \rrbracket \times (\llbracket A \rrbracket \Rightarrow \llbracket B \rrbracket)); ev} \llbracket \Delta \rrbracket \times \llbracket B \rrbracket \\ & \xrightarrow{\llbracket \pi_2 \rrbracket} \llbracket C \rrbracket \end{aligned}$$

– Introduction de 0 à gauche.

$$\llbracket \frac{}{0 \vdash A} 0_i \rrbracket = \llbracket 0 \rrbracket \xrightarrow{i_A} \llbracket A \rrbracket$$

– Introduction de 1 à gauche.

$$\left[\left[\frac{\pi}{\Gamma \vdash A} \right]_{1_r} \right] = [\Gamma, 1] \xrightarrow{\pi_1; [\pi]} [A]$$

– Introduction de 1 à droite.

$$\left[\left[\frac{}{\vdash 1} \right]_{1_r} \right] = [1] \xrightarrow{! [1]} [1]$$

– Introduction du produit à gauche.

$$\left[\left[\frac{\pi}{\Gamma, A, B \vdash C} \right]_{\times_l} \right] = [\Gamma, A \times B] \xrightarrow{\langle \pi_1, \langle \pi_2; \pi_1, \pi_2; \pi_2 \rangle \rangle} [\Gamma, A, B] \xrightarrow{[\pi]} [C]$$

– Introduction du produit à droite.

$$\left[\left[\frac{\pi_1 \quad \pi_2}{\Gamma \vdash A \quad \Gamma \vdash B} \right]_{\times_r} \right] = [\Gamma, \Delta] \xrightarrow{\gamma} [\Gamma] \times [\Delta] \xrightarrow{[\pi_1] \times [\pi_2]} [A \times B]$$

– Première introduction de la somme à droite.

$$\left[\left[\frac{\pi}{\Gamma \vdash A} \right]_{\vdash_r} \right] = [\Gamma] \xrightarrow{[\pi_1]} [A] \xrightarrow{in_1} [A + B]$$

– Seconde introduction de la somme à droite.

$$\left[\left[\frac{\pi}{\Gamma \vdash B} \right]_{\vdash_r} \right] = [\Gamma] \xrightarrow{[\pi_1]} [B] \xrightarrow{in_2} [A + B]$$

– Introduction de la somme à gauche.

$$\begin{aligned} & \left[\left[\frac{\pi_1 \quad \pi_2}{\Gamma, A \vdash C \quad \Delta, B \vdash C} \right]_{\vdash_r} \right] \\ = & [\Gamma, \Delta, A + B] \xrightarrow{\gamma} ([\Gamma] \times [\Delta]) \times ([A] + [B]) \\ & \xrightarrow{[\pi_1 \times [A]; [\pi_1]; \pi_2 \times [B]; [\pi_2]]^{[\Gamma] \times [\Delta]}} [C] \end{aligned}$$

Où $[f|g]^\Gamma$ désigne la construction de co-paire associée aux coproduits faibles fonctoriels de \mathcal{C}_Γ .

– Introduction du μ à gauche.

$$\left[\left[\frac{\pi}{\Gamma, T(A) \vdash A} \right]_{\mu_l} \right] = [\Gamma, \mu X.T] \xrightarrow{[\pi]^\dagger_{[\Gamma]}} [A]$$

Où $[\pi]^\dagger_{[\Gamma]}$ désigne l'action de l'algèbre initiale de $[T]$ dans \mathcal{C}_Γ .

- Introduction du μ à droite.

$$\left[\frac{\frac{\pi}{\Gamma \vdash T(\mu X.T)}}{\Gamma \vdash \mu X.T} \mu_r \right] = \llbracket \Gamma \rrbracket \xrightarrow{\llbracket \pi \rrbracket} \llbracket T(\mu X.T) \rrbracket \xrightarrow{\chi_{\llbracket T \rrbracket}} \llbracket \mu X.T \rrbracket$$

- Introduction du ν à gauche.

$$\left[\frac{\frac{\pi}{\Gamma, T(\nu X.T) \vdash B}}{\Gamma, \nu X.T \vdash B} \nu_l \right] = \llbracket \Gamma, \nu X.T \rrbracket \xrightarrow{\langle \pi_1, \pi_2; \chi_{\llbracket T \rrbracket}^{-1} \rangle} \llbracket \Gamma, T(\nu X.T) \rrbracket \xrightarrow{\llbracket \pi \rrbracket} \llbracket B \rrbracket$$

- Introduction du ν à droite.

$$\left[\frac{\frac{\pi}{\Gamma, A \vdash T(A)}}{\Gamma, A \vdash \nu X.T} \nu_r \right] = \llbracket \Gamma, A \rrbracket \xrightarrow{\llbracket \pi \rrbracket_{\llbracket \Gamma \rrbracket}^{\ddagger}} \llbracket \nu X.T \rrbracket$$

où $\llbracket \pi \rrbracket_{\llbracket \Gamma \rrbracket}^{\ddagger}$ désigne l'action de la coalgèbre terminale de $\llbracket T \rrbracket$ dans \mathcal{C}_{Γ} .

- Foncteur positif.

$$\left[\frac{\frac{\pi}{\Gamma, A \vdash B}}{\Gamma, P(A) \vdash P(B)} [P] \right] = \llbracket \Gamma, P(A) \rrbracket \xrightarrow{\llbracket P \rrbracket_{\llbracket \Gamma \rrbracket}(\llbracket \pi \rrbracket)} \llbracket P(B) \rrbracket$$

- Foncteur négatif.

$$\left[\frac{\frac{\pi}{\Gamma, B \vdash A}}{\Gamma, N(A) \vdash N(B)} [N] \right] = \llbracket \Gamma, N(A) \rrbracket \xrightarrow{\llbracket N \rrbracket_{\llbracket \Gamma \rrbracket}(\llbracket \pi \rrbracket)} \llbracket N(B) \rrbracket$$

Interprétation des règles affaiblies. Notons que si on interprète une preuve π de μLJ affaiblie, le calcul de $\llbracket \pi \rrbracket$ n'utilise pas la structure de types forts de \mathcal{C} puisqu'à aucun moment on a besoin de relever des foncteurs ou des algèbres initiales/coalgèbres terminales. Ainsi pour μLJ affaiblie, l'interprétation des preuves est définie dans n'importe quelle catégorie faiblement μ -fermée. C'est ce qui justifiera *a posteriori* la Proposition 6.1.1 à partir du Théorème 6.4.2 énonçant la validité de l'interprétation ci-dessus.

6.4.2 Correction de l'interprétation

Il faut vérifier que cette interprétation valide les nombreuses règles de réduction de μLJ . Il s'agit à chaque fois de vérifications directes, utilisant la structure qu'on a introduite tout le long de ce chapitre. Passons en revue les différents groupes de règles de réduction de μLJ , et explicitons pour chacun de ces groupes la partie de la structure utilisée.

Coupages logiques. On utilise essentiellement le fait que pour tout $\Gamma \in \mathcal{C}_0$, la catégorie \mathcal{C}_{Γ} est cartésienne fermée et a les coproduits faibles fonctoriels (voir Proposition 1.3.4).

Points fixes. Les équations découlent du fait que $\llbracket \mu X.T \rrbracket$ est interprété comme une algèbre initiale de $\llbracket T \rrbracket_{\llbracket \Gamma \rrbracket}$ dans \mathcal{C}_{Γ} , le cas de $\llbracket \nu X.T \rrbracket$ est identique.

Foncteurs. Pour les constantes et l'identité, c'est évident par définition de l'interprétation. Pour les autres, les équations à montrer sont exactement celles assurées par les Propositions 6.2.7 et 6.3.2.

Coupsures structurelles et axiomes. Calculs immédiats sur la structure cartésienne.

Substitution. La validité de cette règle se montre par l'intermédiaire du lemme suivant :

Lemme 6.4.1. *Soit π_1 une preuve du séquent $\Gamma \vdash A$ et π_2 une preuve de $\Delta, A \vdash B$. Alors :*

$$\llbracket \pi_2[\pi_1/A] \rrbracket = \llbracket \Gamma, \Delta \rrbracket \xrightarrow{\gamma} \llbracket \Delta \rrbracket \times \llbracket \Gamma \rrbracket \xrightarrow{\llbracket \Delta \rrbracket \times \llbracket \pi_1 \rrbracket} \llbracket \Delta \rrbracket \times \llbracket A \rrbracket \xrightarrow{\llbracket \pi_2 \rrbracket} \llbracket B \rrbracket$$

Démonstration. Par induction sur π_2 . Considérons tous les cas de définition de la substitution.

- Règles des groupes axiome et structurel, règles d'introduction de 0, 1, de la flèche et du produit : calcul immédiat sur la structure cartésienne fermée.
- Introductions de la somme à droite : calcul direct sur la structure cartésienne.
- Introduction de la somme à gauche : naturalité de distributivité du produit sur la somme.
- Introduction de μ à droite : calcul direct sur la structure cartésienne.
- Introduction de μ à gauche : calcul direct à partir du Lemme 6.3.3.
- Introduction de ν à gauche : calcul direct sur la structure cartésienne.
- Introduction de ν à droite : calcul direct à partir du Lemme 6.3.4.
- Foncteurs : calcul direct à partir du Lemme 6.3.5.

□

Avec ce lemme, la définition de l'interprétation de la coupure rend immédiate la vérification de la validité de la règle **(Sub)**. On déduit donc de toutes ces vérifications le théorème suivant :

Théorème 6.4.2. *Toute catégorie μ -fermée (pour les formules strictes, pseudo-polynomiales) \mathcal{C} est un modèle correct (du fragment strict, pseudo-polynomial de) μLJ , c'est-à-dire qu'il y a une fonction d'interprétation $\llbracket - \rrbracket$ définie sur les formules et les preuves, et stable par réduction.*

Chapitre 7

Foncteurs ouverts

Sommaire

7.1	Composition verticale	175
7.1.1	Arènes ouvertes	175
7.1.2	Substitution d'arènes	176
7.1.3	Parties fonctorielles	178
7.1.4	Stratégies fonctorielles	181
7.1.5	Restriction verticale	182
7.1.6	Composition verticale	187
7.1.7	Décomposition des stratégies fonctorielles	191
7.2	Foncteurs ouverts et stratégies naturelles	193
7.2.1	Loi d'échange	194
7.2.2	Restriction et composition	196
7.2.3	Foncteurs usuels	199
7.3	Théorème de factorisation	199
7.3.1	Trivialité pour l'identité	200
7.3.2	Factorisation	204
7.4	Transformations dinaturelles ouvertes	209
7.4.1	Échange dinaturel	209
7.4.2	Stabilité par composition	210
7.4.3	Sur la factorisation.	212
7.5	Types Forts	212
7.5.1	Foncteurs ouverts et forces	212
7.5.2	Structure de types forts	214
7.5.3	Foncteur ouverts et types rékursifs	216

On a mentionné au début du Chapitre 5 que μLJ pouvait être vue comme une extension naturelle du μ -calcul propositionnel. Par ailleurs, on sait que les μ -termes peuvent être représentés de façon très précise par des jeux de parité [San02a]. Une des principales caractéristiques des jeux de parité est la représentation des structures de données infinies codées par les μ -termes à l'aide de graphes finis (éventuellement cycliques), construits par induction sur les μ -termes. Le point clé est la représentation des opérateurs de plus petit point fixe et de plus grand point fixe par l'ajout d'une *boucle* dans le graphe. C'est cette représentation des types de données inductifs et coinductifs dans des graphes cycliques qu'on veut généraliser aux jeux

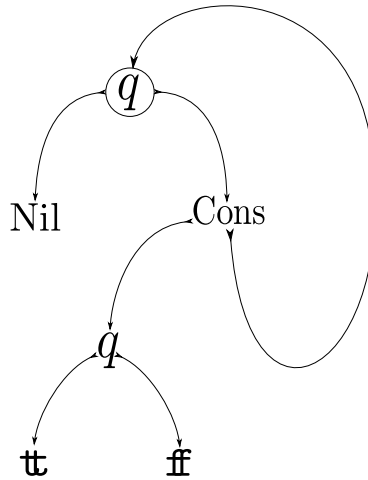


FIG. 7.1 – Représentation des listes de booléens par une arène cyclique.

d'arène : on donne par exemple en Figure 7.1 la représentation désirée des listes (et flots) de booléens dans une arène finie. Outre le caractère élégant de cette représentation, elle permet également de représenter finiment les structures infinies générées par les formules de μLJ , et on a aussi l'espoir d'en tirer ultérieurement des ponts entre les jeux utilisés en vérification et ceux utilisés en sémantique.

Un des principaux problèmes rencontrés dans cette démarche, c'est que la construction de boucle (qu'on introduira formellement au Chapitre 8) est une construction sur les *arènes*, alors qu'on voudra qu'elle donne des algèbres initiales et coalgèbres terminales aux *foncteurs* correspondant à ces arènes. Ainsi, cette volonté de représenter les types inductifs et coinductifs par des boucles impose un travail préliminaire sémantique, consistant à définir des arènes dites *ouvertes* (où certains coups représentent des variables libres) et à associer un *foncteur ouvert* à chacune de ces arènes ouvertes. Le chapitre précédent nous montre qu'il faudra aussi attendre d'autres propriétés de ces foncteurs ouverts, en particulier qu'ils équipent **Inn** et **Par** d'une structure de *types forts*.

La construction des foncteurs ouverts repose sur une idée récurrente en sémantique de jeux : il s'agit de donner un sens général à la notion de stratégie *presque copycat*. Une stratégie presque copycat est une stratégie ayant un comportement potentiellement très complexe sur une partie de l'arène jusqu'à rentrer dans une composante distinguée sur laquelle elle se comporte comme l'identité, se contentant de recopier les coups de son opposant. De telles constructions ont été considérées fréquemment dans la littérature, voir par exemple [AJ94, KNO02]. Dans ce chapitre, on va tenter de donner une définition la plus générale possible de ces stratégies.

Il s'est avéré trop difficile d'effectuer la construction des foncteurs ouverts pour les arènes générales. On a dû imposer la restriction que les *trous* (représentant les variables libres) soient terminaux dans l'arène (c'est à dire qu'ils ne justifient personne), ce qui explique la restriction au fragment pseudo-polynomial de μLJ . Même ainsi, la construction reste très technique et on aimerait, lors de travaux ultérieurs, la simplifier et la généraliser à toutes les arènes. On révèle une structure extrêmement riche sur ces stratégies presque copycat. On commence par définir une notion de *substitution d'arènes*, permettant de donner des expressions de la forme

$S[A]$, où S et A sont des arènes. Par la suite, on porte un intérêt particulier aux arènes de la forme $S[A] \Rightarrow T[B]$, et on dégage deux notions fondamentales :

- La première est celle des *foncteurs ouverts* : étant donné une arène T , le *foncteur ouvert* généré par T associera à toute stratégie $\sigma : A \Rightarrow B$ une stratégie $T(\sigma) : T(A) \rightarrow T(B)$ qu'on contraindra à se comporter comme l'identité sur T et comme σ sur $A \Rightarrow B$.
- La seconde est celle des *transformations naturelles ouvertes* : étant donné une stratégie $\eta : S \Rightarrow T$ satisfaisant des contraintes techniques de *naturalité*, on construira pour toute arène A une stratégie $\eta_A : S[A] \Rightarrow T[A]$ en associant le comportement de η sur $S \Rightarrow T$ et celui de id_A sur $A \Rightarrow A$.

Ces deux notions seront unifiées par une notion de *composition verticale*, permettant de définir une stratégie $\eta[\sigma] : S[A] \Rightarrow T[B]$ à partir de $\eta : S \Rightarrow T$ et $\sigma : A \Rightarrow B$. Par analogie avec les deux notions de composition dans une 2-catégorie on aura bien une loi d'échange entre cette composition verticale et la composition usuelle, justifiant d'un coup la définition de foncteur ouvert et de transformation naturelle ouverte. On montrera également un *théorème de factorisation*, assurant que chaque transformation naturelle entre foncteurs ouverts est une transformation naturelle ouverte. On verra ensuite que la plupart de ces constructions se généralisent au cas dinaturel.

Une fois tous ces outils introduits, il sera facile de montrer que les foncteurs ouverts sont forts, puisque les forces apparaîtront naturellement comme des transformations naturelles ouvertes. On montrera en fait qu'on obtient ainsi une structure de *types forts*, posant la première pierre (et la principale, sur le plan technique) en vue de la construction d'une catégorie μ -fermée de jeux et stratégies totales.

7.1 Composition verticale

7.1.1 Arènes ouvertes

On part des définitions données au Chapitre 2 pour les arènes à adresses explicites. Fixons un ensemble \mathcal{T} de noms de variables. À partir de \mathcal{T} , on considère l'ensemble d'atomes étendu¹ :

$$\Sigma^\square = \Sigma \uplus \{\square_X \mid X \in \mathcal{T}\}$$

Dans ce chapitre et le suivant, les coups dont l'atome est de la forme \square_X seront appelés des *trous* : ils représentent des occurrences de la variable libre X dans l'arène.

Définition 7.1.1. *Une arène ouverte est une arène A bâtie sur l'ensemble d'atomes Σ^\square . On suppose que les trous ne justifient personne dans A , c'est-à-dire que :*

$$\forall m, n \in M_A, m \vdash_A n \Rightarrow \forall X \in \mathcal{T}, \forall \xi \in \mathbb{N}^*, m \neq \xi \cdot \square_X$$

Remarque. Une arène ouverte peut être vue comme une arène usuelle. Dans ce chapitre, toutes les arènes considérées pourront éventuellement être ouvertes, même si ce n'est pas précisé explicitement. On insistera sur le côté ouvert d'une arène uniquement lorsque celui-ci jouera un rôle actif dans la construction courante.

¹Notons que cette définition sous-entend que Σ et $\{\square_X \mid X \in \mathcal{T}\}$ sont disjoints.

Notation. Par abus de notation, on adoptera quand cela ne générera pas d'ambiguïté les conventions suivantes :

- On désignera par \square_X n'importe quel trou t dont l'atome est \square_X . On dira aussi que t est étiqueté par X .
- On désignera par \square^P n'importe quel trou t tel que $\lambda^{OP}(t) = P$, et \square^O n'importe quel trou t tel que $\lambda^{OP}(t) = O$.
- On combinera librement les deux notations ci-dessus, par exemple \square_P^X désignera un trou étiqueté par X et de polarité Joueur.

Si A est une arène ouverte contenant des trous étiquetés par X_1, \dots, X_n , on la notera $A[X_1, \dots, X_n]$, préfigurant ainsi la notation qu'on aura pour le foncteur ouvert associé.

Exemples. Dans une arène ouverte, les trous représentent des variables de types. La Figure 7.2 donne en exemple le graphe de l'arène ouverte correspondant au type $1 + X$, avec X comme variable de type libre.

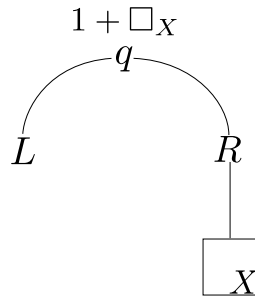


FIG. 7.2 – Arène ouverte pour la formule $1 + X$

Remarque. Les arènes ouvertes peuvent avoir des trous étiquetés par plusieurs variables. Cependant, le plus souvent, on s'intéressera au cas d'une seule variable, qu'on appellera X . Dans tout le développement qui suit, si on ne précise rien, on s'intéressera uniquement aux trous de T étiquetés X .

Définition 7.1.2. Soit T une arène ouverte. On dit que T est de variance mixte si T contient des occurrences de \square_X à la fois en polarité Joueur et Opposant. Sinon, T est de variance fixe. Dans ce cas, on définit sa polarité en disant que T est positive si T ne contient que des occurrences de \square_X^O et négative si elle ne contient que des occurrences de \square_X^P .

7.1.2 Substitution d'arènes

Tout comme pour les formules avec des variables libres, on peut définir pour les arènes ouvertes une opération de substitution, qui consiste à remplacer les trous par une arène complète. Formellement, on définit l'opération comme suit.

Définition 7.1.3 (Substitution d'arènes). Soit $A[X]$ une arène ouverte n'ayant qu'une occurrence de \square_X (c'est-à-dire qu'il existe un unique coup de la forme $\xi \cdot \square_X$ dans M_A) soit B une arène (éventuellement ouverte). On définit la substitution $A[B/X]$, également notée

$A(B)$ quand cela ne cause pas d'ambiguïté, comme suit :

$$\begin{aligned} M_{A[B/X]} &= (M_A \setminus \{\xi \cdot \square_X\}) \uplus \xi \cdot M_B \\ \lambda_{A[B/X]} &= \begin{cases} [\lambda_{A \setminus \xi \cdot \square_X}, \xi \cdot \lambda_B] & \text{si } \lambda_A(\xi \cdot \square_X) = O \\ [\lambda_{A \setminus \xi \cdot \square_X}, \xi \cdot \overline{\lambda_B}] & \text{sinon} \end{cases} \\ m \vdash_{A[B]} n &\Leftrightarrow \vee \begin{cases} m \vdash_A n \\ m (\xi \cdot \vdash_B) n \\ m \vdash_A \xi \cdot \square_X \wedge n \in I_B \end{cases} \end{aligned}$$

Il est évident que les éventuels trous sont toujours terminaux dans $A[B]$, le résultat de l'opération de substitution est donc une arène ouverte bien définie.

Substitutions successives. Il nous faut étudier quelques propriétés des substitutions successives. La première raison est de montrer comment, à partir de la définition ci-dessus, généraliser la substitution d'arènes au cas où on a plusieurs occurrences du trou dans l'arène ouverte. L'autre raison est qu'on va être amenés plus tard à considérer des foncteurs ouverts d'arité supérieure, qui correspondent à des arènes ouvertes avec plusieurs types de trous, par exemple étiquetés X et Y . Pour donner sens à la substitution simultanée de \square_X et \square_Y , la situation est similaire au cas de la substitution dans les termes : il nous faut montrer un lemme de commutation.

Lemme 7.1.1 (Substitutions successives indépendantes). *Si T contient une occurrence de \square_X et une occurrence de \square_Y , et si A et B sont des arènes ne contenant ni \square_X ni \square_Y :*

$$T[A/X][B/Y] = T[B/Y][A/X]$$

Ainsi, on la notera $T[A/X, B/Y]$ sans ambiguïté. De même, si T contient plusieurs occurrences de \square_X , on définit $T[A/X]$ par substitutions successives (en renommant au préalable les X en X_1, \dots, X_n distincts). Par l'égalité ci-dessus, l'ordre de substitution n'importe pas et le résultat est bien défini.

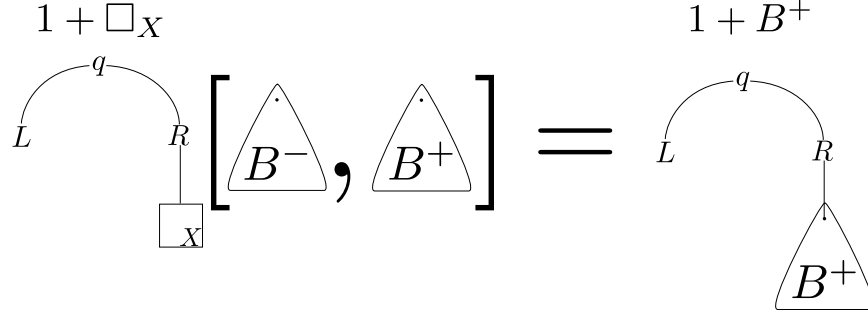
Démonstration. Immédiat par définition de la substitution. □

Remarque. C'est ici qu'il est crucial que les arènes soient *localisées*, c'est-à-dire qu'on ait la maîtrise des adresses. Sinon, on serait forcé de définir les coups de $A[B/X]$ par :

$$M_{A[B/X]} = M_A \setminus \{\square_X\} + M_B$$

et on perdrait le lemme précédent, parce que l'opération de somme disjointe sur les ensembles n'est commutative qu'à isomorphisme près. Pire encore, on perdrait pour la même raison des équations fort naturelles, comme $(A \times \square_X)[B/X] = A \times B$. L'autre solution serait de quotienter les arènes par isomorphisme d'arènes, mais on perdrait alors tout l'intérêt de la représentation des points fixes par des arènes finies cycliques, car on ne distinguerait plus une arène finie de son expansion infinie.

Notation. Soit S une arène ouverte, contenant éventuellement des occurrences Joueur et Opposant de \square_X . Alors si A et B sont des arènes, on note $S[A/X^-, B/X^+]$ ou simplement $S[A, B]$ l'arène obtenue par substitutions successives des occurrences de \square_X^P par A et des occurrences de \square_X^O par B .

FIG. 7.3 – Substitution $(1 + X)[B^-, B^+]$.

Exemple. On peut observer dans la Figure 7.3 la représentation en graphe du résultat de la substitution de \square_X^P par B^+ et de \square_X^Q par B^- dans l'arène correspondant à la formule $1 + X$.

Lemme 7.1.2 (Substitutions successives en cascade). *Soient S, T deux arènes ouvertes et A une arène. On a l'égalité suivante :*

$$(S[T/X])[A/X] = S[T[A/X]/X]$$

Démonstration. Immédiat. □

7.1.3 Parties fonctorielles

Soient $S[X]$ et $T[X]$ des arènes ouvertes. Dans cette sous-section, on va s'intéresser aux arènes de la forme $S[A, B] \Rightarrow T[C, D]$ (A remplace les occurrences de \square_X^P dans S , B remplace les occurrences de \square_X^Q dans S , ainsi de suite pour T). Plus particulièrement, on va étudier les parties légales sur ce genre d'arène et tenter de repérer dans quels cas ces parties sont «fonctorielles», c'est-à-dire que les parties sur $S[X] \Rightarrow T[X]$ et celles sur A, B, C, D sont indépendantes et peuvent être séparées.

Terminologie. Soit $s \in \mathcal{L}_{S[A, B] \Rightarrow T[C, D]}$. Si $X \in \{A, B, C, D\}$, une *copie* de X dans s est la donnée d'une adresse ξ telle que $\xi \cdot \square_X$ est substitué par X dans $S[A, B] \Rightarrow T[C, D]$, et d'un pointeur vers un coup s_j . Ainsi, chaque coup de X dans s est dans une unique copie (ξ, j) de X . Si s_i est un coup initial de X dans s , on dit que s_i *ouvre* une copie (ξ, j) de X si ξ est préfixe de l'adresse de s_i et si s_i pointe sur s_j . Par extension, on dit qu'une copie (ξ, j) de X est *ouverte* dans s s'il existe un coup initial qui l'a ouverte.

Remarque. Intuitivement, tous les coups contenus dans une copie (ξ, j) de X correspondent au même trou \square_Y . On peut voir l'opération de restriction fonctorielle (à venir) comme une opération d'«agglomération» de tous ces coups.

Définition 7.1.4 (Composantes). *Considérons une arène $E = S[A, B] \Rightarrow T[C, D]$, soit $m \in M_E$. On dit que m est dans la composante fonctorielle de E si m est dans S ou T . De même, on dit que m est dans la composante argument s'il est dans A, B, C ou D .*

Définition 7.1.5 (Saut). Soit $s \in \mathcal{L}_{S[A,B] \Rightarrow T[C,D]}$. Considérons un préfixe $s'ab \sqsubseteq s$. On dit que b est un saut si l'on est dans une des situations suivantes (dans cette définition, les polarités (Opposant, Joueur) sont globales, c'est-à-dire relatives à l'arène $S[A, B] \Rightarrow T[C, D]$ et pas à A, B, C ou D).

1. a est un coup Opposant dans la composante foncteur, et b est un coup Joueur dans la composante argument ;
2. a est un coup Joueur dans la composante foncteur, et b est un coup Opposant non-initial dans la composante argument ;
3. a est un coup dans la composante argument, et b est dans la composante foncteur ;
4. b est un coup initial de la composante argument, sachant qu'il y avait déjà un coup initial i de la composante argument de même polarité que b dans sa , et i et b sont dans des lieux différents : soit ils ouvrent deux sous-arènes $X, Y \in \{A, B, C, D\}$ différentes, soit ils ouvrent deux copies différentes $(\xi', j') \neq (\xi, j)$ d'une même sous-arène $X \in \{A, B, C, D\}$.

On dit que b est un saut Opposant (resp. saut Joueur) si $\lambda_{S[A,B] \Rightarrow T[C,D]}^{OP}(b) = O$ (resp. $\lambda_{S[A,B] \Rightarrow T[C,D]}^{OP}(b) = P$).

Remarque. La notion de saut est relative à une arène, et à un choix de construction de cette arène par substitution ! Par exemple si on avait $S'[A', B'] \Rightarrow T'[C', D'] = S[A, B] \Rightarrow T[C, D]$, les notions de saut ne coïncideraient pas nécessairement pour les deux constructions, comme en témoignera l'exemple à venir. Les notions de sauts ainsi que les définitions subséquentes devront donc être utilisées seulement en présence d'une décomposition explicite de l'arène par substitution. Notons que lorsqu'on donne une arène sous la forme $S[A, B] \Rightarrow T[C, D]$, c'est évidemment cette construction qui définit la notion de saut utilisée.

Remarque. Interdire les sauts permettra de contraindre suffisamment la forme des parties pour qu'on puisse aisément en calculer la restriction aux composantes fonctorielle et argument. En particulier, la définition de saut de type 4 mérite une petite explication. Imaginons qu'on ait une partie s , et qu'on veuille en calculer la restriction à la composante fonctorielle. Cette restriction fonctorielle conserve une réminiscence de la dynamique de s sur la composante argument : on doit reconstruire des occurrences des trous, en se basant sur les copies (ξ, j) de sous-arènes $X \in \{A, B, C, D\}$ visitées dans s . Mais pour écrire précisément ces trous, il faut deux choses : une adresse et un pointeur. Il est donc nécessaire de pouvoir dégager une adresse et un pointeur uniques à partir de tous les coups initiaux arguments de même polarité dans s , il faut donc qu'ils partagent ces mêmes adresses et pointeurs, d'où la définition de saut de type 4.

Exemple. Observons la partie légale décrite en Figure 7.4. Alors s ne comporte pas de saut pour la décomposition $(1 + X)[\mathbf{nat}] \Rightarrow (1 + \mathbf{nat} \times X)[\mathbf{bool}]$. Cependant, elle en comporte un pour $(1 + X)[\mathbf{nat}] \Rightarrow (1 + X \times \mathbf{bool})[\mathbf{nat}]$. En effet, pour cette seconde décomposition, $q_{\mathbf{nat}}$ est un saut de type 1 : $q_{\mathbf{bool}}$ est un coup Opposant dans la composante foncteur, et $q_{\mathbf{nat}}$ est un coup Joueur dans la composante argument.

Définition 7.1.6 (Parties fonctorielles). Une partie $s \in \mathcal{L}_{S[A,B] \Rightarrow T[C,D]}$ est dite O -fonctorielle s'il n'y a pas de saut Opposant dans s . De même, elle est dite P -fonctorielle s'il n'y a pas de saut Joueur dans s . Elle est bi-fonctorielle s'il n'y a ni saut Joueur ni saut Opposant dans s .

Lemme 7.1.3. Toute P -vue s sur $\mathcal{L}_{S[A,B] \Rightarrow T[C,D]}$ est O -fonctorielle.

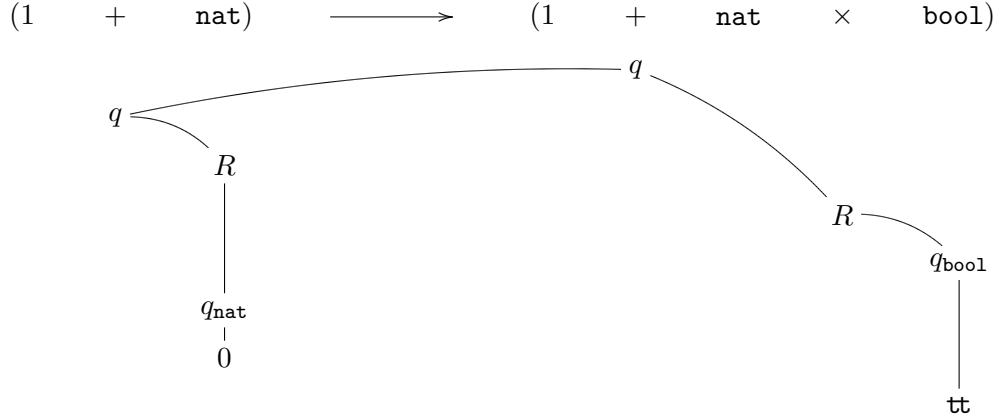


FIG. 7.4 – Exemple de partie fonctorielle.

Démonstration. Par induction sur s . À chaque étape, Opposant pointe vers le coup précédent. Si ce coup est dans la composante foncteur, Opposant jouera nécessairement soit un coup dans la composante foncteur (qui ne peut être un saut), soit un coup initial dans la composante argument (qui ne peut pas plus être un saut, la forme des P -vues interdit le cas 4). De même, si le coup précédent est un coup interne de la composante argument, Opposant pointera vers ce coup, donc jouera nécessairement un coup de la composante argument qui ne pourra pas être un saut. \square

Nous allons examiner les parties bi-fonctorielles sur une arène $S[A, B] \Rightarrow T[C, D]$ et montrer qu'on peut les caractériser par une forme particulière.

Lemme 7.1.4 (Forme canonique). *Soit $s \in \mathcal{L}_{S[A, B] \Rightarrow T[C, D]}$. Alors, s est bi-fonctorielle si et seulement si s se décompose en $s_1 s_2$ avec s_1 uniquement dans la composante foncteur et s_2 uniquement dans la composante argument, de sorte que tous les coups initiaux Opposants de la composante argument dans s_2 ouvrent une même copie (ξ, j) de $X \in \{A, D\}$ et que tous les coups initiaux Joueur de la composante argument dans s_2 ouvrent une même copie (ξ, j) de $X \in \{B, C\}$.*

Démonstration. Montrons le sens direct par induction sur s . Si s est vide c'est évident. Sinon, $s = s'a$. Par hypothèse d'induction, $s' = s'_1 s'_2$. Plusieurs cas se présentent alors :

- Soit s'_2 est vide, auquel cas s' est entièrement dans la composante foncteur. Si a est dans la composante argument, la décomposition $s = s'a$ vérifie bien les conditions requises. Par ailleurs, si a est dans la composante foncteur, la décomposition $s = (s'a)\epsilon$ vérifie également ces conditions.
- Soit s'_2 n'est pas vide. Dans ce cas, a doit également être dans la composante argument, sinon ce serait un saut. De plus, si c'est un coup initial dans $X \in \{A, B, C, D\}$, il doit partager le même pointeur et la même adresse que les coups initiaux de la composante argument précédents et de même polarité, sinon ce serait un saut.

Réciproquement, il est clair qu'une partie munie d'une telle décomposition ne comporte pas de saut. \square

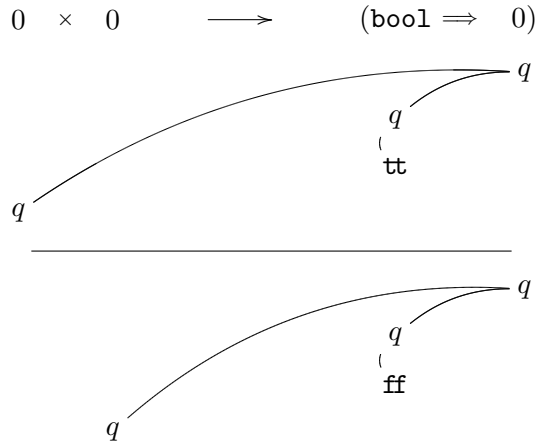


FIG. 7.5 – P -vues bi-fonctorielles sur $(X \times X)[0] \Rightarrow X[\text{bool} \Rightarrow 0]$, formant une stratégie non fonctorielle.

Exemple. Dans l'exemple de la Figure 7.4, la partie légale s est bi-fonctorielle pour la décomposition $(1 + X)[\text{nat}] \Rightarrow (1 + \text{nat} \times X)[\text{bool}]$, mais pas pour $(1 + X)[\text{nat}] \Rightarrow (1 + X \times \text{bool})[\text{nat}]$.

7.1.4 Stratégies fonctorielles

On a défini les parties P -fonctorielles comme celles où Joueur n'effectue jamais de saut entre les deux composantes. Comme d'habitude lorsqu'on définit une sous-classe des parties légales, il est intéressant d'examiner la classe de stratégies correspondante. Dans le cas présent, ces stratégies sont celles qui entretiennent une certaine isolation entre les composantes fonctorielle et argument : bien qu'elles puissent encore dépendre de l'histoire de la partie sur les deux composantes, elles ne peuvent pas effectuer de sauts de l'une à l'autre.

Définition 7.1.7. *Considérons une arène $S[A, B] \Rightarrow T[C, D]$. Une stratégie $\sigma : S[A, B] \Rightarrow T[C, D]$ est fonctorielle si elle est innocente et si toute partie $s \in \sigma$, si s est O -fonctorielle, alors s est bi-fonctorielle.*

Il aurait été tentant de définir les stratégies fonctorielles comme celles dont les parties sont P -fonctorielles, mais un saut Opposant peut très bien provoquer un saut Joueur, entraînant une rupture de la P -fonctorialité. De même, il est tout à fait possible que toutes les P -vues d'une stratégie σ soient fonctorielles et que σ ne soit pas fonctorielle pour autant, on peut observer un exemple sur la Figure 7.5. On pourrait définir cette notion dans un cadre plus général que celui des stratégies innocentes, mais cela crée des complications techniques alors que dans les cas qui nous intéressent ici, les stratégies considérées seront toujours innocentes. Montrons maintenant quelques propriétés sur ces stratégies fonctorielles et sur leur composition.

Lemme 7.1.5. *Soient $\sigma, \tau : S[A, B] \Rightarrow T[C, D]$ deux stratégies fonctorielles. Si elles ont les mêmes parties bi-fonctorielles, alors elles sont égales.*

Démonstration. Considérons leurs P -vues, par définition des stratégies fonctorielles elles sont P -fonctorielles. D'autre part, le Lemme 7.1.3 assure qu'elles sont O -fonctorielles. Or, si σ et

τ ont les mêmes P -vues, elles sont égales puisqu'elles sont innocentes. Il suffit donc de les comparer sur les parties bi-fonctorielles. \square

Ainsi, les stratégies fonctorielles sont caractérisées par leurs parties bi-fonctorielles. C'est intéressant, car seules ces parties bi-fonctorielles vont pouvoir être décomposées en une partie sur la composante fonctorielle et une partie sur la composante argument. Pour montrer les propriétés de ces stratégies bi-fonctorielles vis-à-vis de la composition (notamment celles qui seront introduites plus tard par composition verticale), on va montrer qu'il suffit de considérer les interactions «bi-fonctorielles», au sens où toutes leurs projections sont bi-fonctorielles.

Définition 7.1.8. *Soit $u \in I(S[A, B], T[C, D], U[E, F])$ une interaction. Un saut sur u est un coup u_i qui est un saut sur une des restrictions $u_{\upharpoonright S[A, B], U[E, F]}$, $u_{\upharpoonright S[A, B], T[C, D]}$ ou $u_{\upharpoonright T[C, D], U[E, F]}$. Par extension, on dit que u est bi-fonctorielle si elle ne comporte pas de saut.*

Cette notion de bi-fonctorialité sur les interactions permet de démontrer la stabilité par composition des stratégies fonctorielles.

Lemme 7.1.6. *Soit $\sigma : S[A, B] \Rightarrow T[C, D]$ fonctorielle et $\tau : T[C, D] \Rightarrow U[E, F]$ fonctorielle, alors $\sigma; \tau : S[A, B] \Rightarrow U[E, F]$ est fonctorielle.*

Démonstration. Soit $u \in \sigma \parallel \tau$, telle que $u_{\upharpoonright S[A, B], U[E, F]}$ est O -fonctorielle. Considérons par l'absurde un éventuel premier saut dans u . Ce premier saut correspondrait alors soit à un coup de σ , soit à un coup de τ (contredisant ainsi leur fonctorialité, puisque c'est le *premier* saut), soit à un coup de l'Opposant externe. Si c'est un saut sur $u_{\upharpoonright S[A, B], U[E, F]}$, on contredit immédiatement l'hypothèse de O -fonctorialité de $u_{\upharpoonright S[A, B], U[E, F]}$. Si c'est un saut sur $u_{\upharpoonright S[A, B], T[C, D]}$ ou $u_{\upharpoonright T[C, D], U[E, F]}$, un raisonnement direct par cas sur les types de sauts montre que c'est également un saut dans $u_{\upharpoonright S[A, B] \Rightarrow U[E, F]}$ ce qui contredit l'hypothèse de O -fonctorialité. Ainsi, $u_{\upharpoonright T[C, D], U[E, F]}$ est également P -fonctorielle. \square

7.1.5 Restriction verticale

Dans cette sous-section, on ne considèrera plus que des arènes de la forme $S[B, A] \Rightarrow T[A, B]$. Ce cas s'avèrera en effet suffisamment général pour définir les foncteurs ouverts et les transformations (di)naturelles ouvertes, et permet une certaine simplification technique : si $s \in \mathcal{L}_{S[B, A] \Rightarrow T[A, B]}$ est bi-fonctorielle, on va pouvoir définir sa restriction verticale sur $A \Rightarrow B$ directement. Pour $S[A, B] \Rightarrow T[C, D]$, il aurait fallu distinguer les cas selon les sous-arènes visitées parmi A, B, C et D .

Restriction fonctorielle. Il y a deux notions de restriction verticale : celle à la composante fonctorielle, et celle à la composante argument. On va d'abord donner la restriction fonctorielle et prouver qu'elle est correcte (*i.e.* qu'elle construit des parties légales et qu'elle vérifie certaines propriétés vis-à-vis des interactions), puis on fera de même pour la restriction argument.

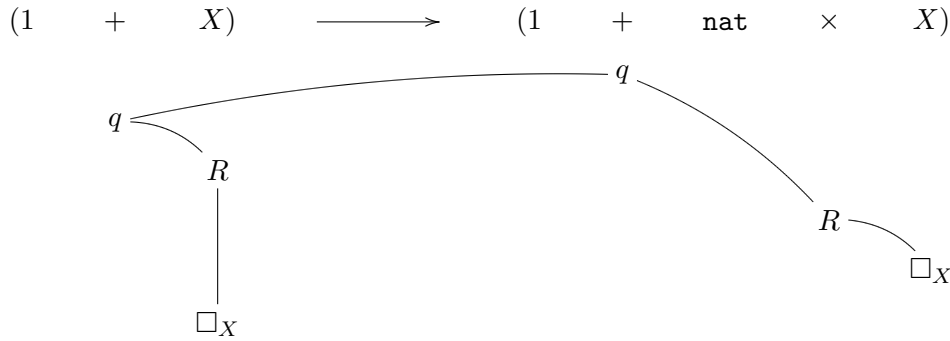
Définition 7.1.9. *Soit $s \in \mathcal{L}_{S[B, A] \Rightarrow T[A, B]}$ une partie bi-fonctorielle. On définit sa restriction fonctorielle $s_{\upharpoonright S[X] \Rightarrow T[X]}$ par :*

- s si s ne joue que dans la composante fonctorielle ;
- $s_{\upharpoonright S, T} \square_X$ si s n'atteint que B , avec le pointeur et l'adresse correspondant à ceux des coups initiaux dans B (choix unique par le Lemme 7.1.4) ;
- $s_{\upharpoonright S, T} \square_X \square_X$ si s atteint B et A , avec les pointeurs et adresses choisis comme ci-dessus, respectivement pour B puis A .

Proposition 7.1.7. *Soit $s \in \mathcal{L}_{S[B,A] \Rightarrow T[A,B]}$ une partie bi-fonctorielle. Alors, $s_{\upharpoonright S[X] \Rightarrow T[X]} \in \mathcal{L}_{S \Rightarrow T}$ est bien une partie légale.*

Démonstration. Immédiat par la décomposition du Lemme 7.1.4. \square

Exemple. Considérons la partie de la Figure 7.4, sur la décomposition de l'arène $(1 + X)[\mathbf{nat}] \Rightarrow (1 + \mathbf{nat} \times X)[\mathbf{bool}]$. Comme on l'a vu, cette partie est bi-fonctorielle. On peut donc extraire sa *restriction fonctorielle* $s_{\upharpoonright (1+X) \Rightarrow (1+\mathbf{nat} \times X)}$:



Notons que si Joueur avait immédiatement répondu à $q_{\mathbf{bool}}$ sans passer par \mathbf{nat} , la restriction fonctorielle se serait arrêtée un coup avant : elle aurait donc eu une longueur impaire et aurait fini par un coup Opposant.

Lemme 7.1.8. *Supposons que les trous ont la même polarité dans S, T et U . Soit $u \in I(S[A], T[B], U[C])$ bi-fonctorielle, alors on peut en extraire $\text{Fonc}(u) \in I(S, T, U)$ telle que :*

$$\begin{aligned}
 \text{Fonc}(u)_{\upharpoonright S, T} &= (u_{\upharpoonright S[A], T[B]})_{\upharpoonright S[X] \Rightarrow T[X]} \\
 \text{Fonc}(u)_{\upharpoonright T, U} &= (u_{\upharpoonright T[B], U[C]})_{\upharpoonright T[X] \Rightarrow U[X]} \\
 \text{Fonc}(u)_{\upharpoonright S, U} &= (u_{\upharpoonright S[A], U[C]})_{\upharpoonright S[X] \Rightarrow U[X]}
 \end{aligned}$$

Démonstration. L'interaction u étant bi-fonctorielle, par application du Lemme 7.1.4 sur ses composantes, elle est de la forme $u_1 u_2$ avec u_1 entièrement incluse dans la composante foncteur et u_2 entièrement incluse dans la composante argument. Notons également que par le Lemme 7.1.9, pour chaque composante $X \in \{A, B, C\}$, tous les coups initiaux de X pointent vers le même coup et partagent la même adresse. Notons que les arènes de la composante foncteur sont nécessairement visitées dans l'ordre A, B, C ou C, B, A . Montrons-le dans le cas où \square_X est un coup Opposant dans S, T et U : par bi-fonctorialité, seul Opposant peut inaugurer la composante argument en jouant sur C . Mais tout coup initial de A est Joueur dans $S[A] \Rightarrow T[B]$, donc par P -fonctorialité il doit suivre un coup de la composante argument, qui ne peut être que dans B (puisque A n'est pas encore ouvert). Le même raisonnement tient pour B , ainsi ces sous-arènes sont ouvertes dans l'ordre C, B, A . Donnons maintenant la construction de $\text{Fonc}(u)$, selon les cas suivants :

- Si u n'atteint ni A , ni B , ni C , on a $u = u_1$. On définit simplement $\text{Fonc}(u) = u$. Les égalités à montrer sont triviales puisque les restrictions fonctorielles se comportent ici comme l'identité.

- Si u n'atteint que C (resp. A si les trous apparaissent en polarité Joueur dans S, T et U), alors tous les coups initiaux de C (resp. A) pointent vers le même coup dans u_1 . On pose alors $Fonc(u) = u_1 \square_X$, où \square_X pointe vers le justificateur commun des coups initiaux de C (resp. A). Les égalités sont évidentes par définition de la restriction verticale.
- Si u atteint C (resp. A) et B , on pose $Fonc(u) = u_1 \square_X \square_X$, où la première occurrence de \square_X pointe comme les coups initiaux de C (resp. A) et la seconde comme les coups initiaux de B . Les égalités sont évidentes par définition de la restriction verticale.
- Si u atteint C, B et A dans cet ordre (resp. A, B et C), on pose $Fonc(u) = u_1 \square_X \square_X \square_X$, où les trois occurrences de \square_X pointent comme les coups initiaux de C, B et A (resp. A, B et C). Les égalités sont évidentes.

□

Restriction argument. Passons maintenant à la partie argument de la restriction verticale. Sa définition repose sur une analyse plus approfondie de la décomposition du Lemme 7.1.4. Partons de cette décomposition, et examinons maintenant la structure de la partie argument obtenue.

Lemme 7.1.9. *Considérons $s \in \mathcal{L}_{S[B,A] \Rightarrow T[A,B]}$ bi-fonctorielle, et sa décomposition $s_1 s_2$ donnée par le Lemme 7.1.4. s_2 vérifie alors la propriété suivante : si $s'_2 \sqsubseteq s_2$ finit par un coup initial de A , il existe un unique coup initial de B dans $\lceil s_1 s'_2 \rceil$ qu'on appelle son candidat de justification.*

Démonstration. Commençons par noter que $\lceil s_1 s'_2 \rceil$ ne peut contenir au plus qu'un seul coup initial dans B . En effet, un coup initial dans B ne peut pointer que dans la composante foncteur par la structure des arènes ouvertes, et c'est un coup Opposant par hypothèse. Donc, si on rencontre un tel coup lors du calcul de la P -vue, on va ensuite suivre son pointeur et arriver dans la composante foncteur, donc dans s_1 . Comme s_1 est entièrement inclus dans la composante foncteur, il est clair qu'on y croisera pas d'autre occurrence d'un coup initial sur B . Donc, on a au plus un tel coup initial dans la vue.

De même, on en a au moins un : considérons tous les coups initiaux sur A apparaissant dans $\lceil s_1 s'_2 \rceil$. Prenons le premier, et examinons le coup précédent. Ça ne peut pas être un coup sur A , puisqu'on a supposé qu'on avait le premier coup initial sur A . Par ailleurs, ce coup est nécessairement dans la composante argument, sinon ce premier coup initial sur A serait un saut pour Joueur. Donc c'est un coup sur B , et le calcul de la P -vue à partir de ce coup atteint nécessairement un coup initial sur B . □

Le lemme précédent ouvre désormais la voie à la définition de la restriction verticale à la partie argument, comme suit :

Définition 7.1.10. *Soit $s \in \mathcal{L}_{S[B,A] \Rightarrow T[A,B]}$ bi-fonctorielle. Par le Lemme 7.1.4, s est de la forme $s_1 s_2$, avec s_1 dans la composante foncteur et s_2 dans la composante argument. On définit alors $s \upharpoonright_{A \Rightarrow B} = s_2$, où chaque coup initial de A pointe vers son unique candidat de justification, défini par le Lemme 7.1.9.*

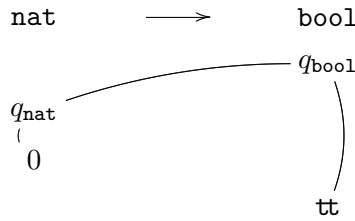
Remarque. Formellement, la définition précédente nécessite une phase de réadressage : on doit ôter de chaque coup le préfixe de son adresse correspondant à la substitution de A, B, C, D dans S et T . Le lecteur familier des jeux d'arènes remarquera l'analogie entre cette construction et celle de l'action bifonctorielle du connecteur \Rightarrow . Si $\sigma : A \Rightarrow B$ et $\tau : C \Rightarrow D$, on définit

$\sigma \Rightarrow \tau : (B \Rightarrow C) \Rightarrow (A \Rightarrow D)$ par l'ensemble des parties s telles que $s_{\downarrow C,D} \in \tau$ (pas de problème), et on voudrait écrire $s_{\downarrow A,B} \in \sigma$ mais c'est impossible : cette restriction perd des pointeurs. Il faut alors mettre la partie sous la forme d'une interaction sur $I(A, B, C, D)$ pour faire apparaître les «candidats de justification» et déterminer les pointeurs des coups initiaux de A .

Proposition 7.1.10. *Soit $s \in \mathcal{L}_{S[B,A] \Rightarrow T[A,B]}$ bi-fonctorielle, alors $s_{\downarrow A \Rightarrow B} \in \mathcal{L}_{A \Rightarrow B}$.*

Démonstration. Immédiat par construction. □

Exemple. Considérons la partie de la Figure 7.4, sur la décomposition de l'arène $(1 + X)[\text{nat}] \Rightarrow (1 + \text{nat} \times X)[\text{bool}]$. Sa restriction argument $s_{\downarrow \text{nat} \Rightarrow \text{bool}}$ est donnée par le diagramme suivant :



On garde tous les pointeurs comme dans s , sauf pour q_{nat} qu'on fait pointer vers l'unique coup initial de bool dans sa P -vue.

Lemme 7.1.11. *Supposons que les trous ont la même polarité dans S, T et U . Soit $u \in I(S[A], T[B], U[C])$ bi-fonctorielle, P -visible sur $S[A] \Rightarrow T[B]$ et $T[B] \Rightarrow U[C]$. Alors on peut en extraire $Arg(u) \in I(A, B, C)$ (resp. $I(C, B, A)$ si les trous ont polarité Joueur dans S, T et U) telle que :*

$$\begin{aligned} Arg(u)_{\downarrow A, B} &= (u_{\downarrow S[A], T[B]})_{\downarrow A \Rightarrow B} \\ Arg(u)_{\downarrow B, C} &= (u_{\downarrow T[B], U[C]})_{\downarrow B \Rightarrow C} \\ Arg(u)_{\downarrow A, C} &= (u_{\downarrow S[A], U[C]})_{\downarrow A \Rightarrow C} \end{aligned}$$

Démonstration. On va faire la preuve dans le cas où les trous apparaissent en polarité Opposant dans S, T et U , l'autre cas étant similaire. On construit $Arg(u)$ comme suit : par le Lemme 7.1.4, u se décompose en $u_1 u_2$, où u_1 est entièrement inclus dans la composante fonctorielle et u_2 dans la composante argument. Pour obtenir $Arg(u)$, on conserve les coups de u_2 et les pointeurs, mais on doit définir de nouveaux pointeurs pour les coups initiaux de A et B . Si dans $u'a \sqsubseteq u_2$, a est un coup initial sur A , on définit son nouveau pointeur comme l'unique candidat de justification donné par le Lemme 7.1.9 sur A, B . On fait ensuite de même pour les coups initiaux sur B . Les égalités à démontrer sont évidentes sur les coups et les pointeurs des coups internes. Examinons la situation des pointeurs pour les coups initiaux. Les deux premières égalités restent évidentes puisque les définitions des pointeurs des coups initiaux coïncident. Pour la troisième égalité, à gauche, un coup initial de A pointera sur un coup initial de C s'il pointait dans $Arg(u)$ sur un coup initial de B qui pointait sur un coup initial de B sachant que ces deux pointeurs étaient définis par le Lemme 7.1.9. À droite, un coup initial i_A de A pointe vers un coup initial i_C de C si i_C est l'unique coup initial de C visible

par i_A dans $S[A] \Rightarrow U[C]$. Or, par le Lemme 2.2.7 des préliminaires, s'il y a i_B tel que i_A voit i_B dans $S[A] \Rightarrow T[B]$ et i_C tel que i_B voit i_C dans $T[B] \Rightarrow U[C]$, alors i_A voit i_C dans $S[A] \Rightarrow U[C]$, donc i_A doit bien pointer vers i_C dans $Arg(u)_{\downarrow A \Rightarrow C}$ par unicité d'un tel i_C . \square

Compatibilité avec les P -vues. On va maintenant montrer que le calcul des différentes restrictions commute avec le calcul de la P -vue par l'intermédiaire du lemme suivant, qui s'avèrera très utile pour montrer les propriétés relatives à l'innocence de la composition verticale.

Lemme 7.1.12 (Compatibilité). *Soit $s \in \mathcal{L}_{S[B,A] \Rightarrow T[A,B]}$ une partie bi-fonctorielle. On a les égalités :*

$$\begin{aligned} \lceil s \downarrow_{S[X] \Rightarrow T[X]} \rceil &= (\lceil s \rceil) \downarrow_{S[X] \Rightarrow T[X]} \\ \lceil s \downarrow_{A \Rightarrow B} \rceil &= (\lceil s \rceil) \downarrow_{A \Rightarrow B} \end{aligned}$$

Démonstration. Immédiat par induction sur le calcul de la P -vue et le Lemme 7.1.4. \square

Restrictions successives. Pour construire les foncteurs ouverts d'arité multiple, il faut s'intéresser au comportement de la restriction verticale en présence de trous étiquetés par plusieurs variables libres X et Y .

Lemme 7.1.13. *Soient S, T des arènes ouvertes avec des trous étiquetés X et Y . Soit $s \in \mathcal{L}_{S[B/X^-, A/X^+][D/Y^-, C/Y^+] \Rightarrow T[A/X^-, B/X^+][C/Y^-, D/Y^+]}$ une partie bi-fonctorielle en X et Y . Alors, les égalités suivantes sont satisfaites :*

$$\begin{aligned} (s \downarrow_{S[B/X^-, A/X^+][Y] \Rightarrow T[A/X^-, B/X^+][Y]}) \downarrow_{A \Rightarrow B} &= s \downarrow_{A \Rightarrow B} \\ (s \downarrow_{[D/Y^-, C/Y^+][X] \Rightarrow T[C/Y^-, D/Y^+][X]}) \downarrow_{C \Rightarrow D} &= s \downarrow_{C \Rightarrow D} \\ (s \downarrow_{S[B/X^-, A/X^+][Y] \Rightarrow T[A/X^-, B/X^+][Y]}) S[X] \Rightarrow T[X] &= \\ (s \downarrow_{S[D/Y^-, C/Y^+][X] \Rightarrow T[C/Y^-, D/Y^+][X]}) \downarrow_{S[Y] \Rightarrow T[Y]} & \end{aligned}$$

Démonstration. Il suffit de remarquer qu'une partie bi-fonctorielle en X et Y ne pourra atteindre qu'un seul de $A \Rightarrow B$ ou $C \Rightarrow D$, par le Lemme 7.1.4, remarque qui trivialisait les égalités à démontrer. \square

De même, pour étudier les propriétés des compositions verticales successives, il faut examiner le comportement de la composition verticale sur des arènes ouvertes obtenues par substitutions en cascade.

Lemme 7.1.14. *Soient S, T, U, V des arènes ouvertes, soit $s \in \mathcal{L}_{S[U/X][A/X] \Rightarrow T[V/X][B/X]}$ bi-fonctorielle pour $S \Rightarrow T$ et pour $S[U/X] \Rightarrow T[V/X]$. Alors, on a les égalités suivantes :*

$$\begin{aligned} (s \downarrow_{(S[U/X])[X] \Rightarrow (T[V/X])[X]}) \downarrow_{S[X] \Rightarrow T[X]} &= s \downarrow_{S[X] \Rightarrow T[X]} \\ (s \downarrow_{(S[U/X])[X] \Rightarrow (T[V/X])[X]}) \downarrow_{U \Rightarrow V} &= (s \downarrow_{U[A/X] \Rightarrow V[B/X]}) \downarrow_{U[X] \Rightarrow V[X]} \\ s \downarrow_{A \Rightarrow B} &= (s \downarrow_{U[A/X] \Rightarrow V[B/X]}) \downarrow_{A \Rightarrow B} \end{aligned}$$

Démonstration. Les termes de gauche sont valides dans $\mathcal{L}_{(S[T/X])[A/X] \Rightarrow U[V/X][B/X]}$, ceux de droites sont valides dans $\mathcal{L}_{S[T[A/X]/X] \Rightarrow U[V[B/X]/X]}$. Cependant, ces deux ensembles sont égaux par le Lemme 7.1.2. Deux applications du Lemme 7.1.4 montrent que les parties bi-fonctorielles dans cet ensemble ont pour forme $s_1 s_2 s_3$, avec s_1 dans S, U , s_2 dans T, V et s_3 dans A, B . Le même Lemme 7.1.4 montre également que les pointeurs des coups initiaux de chaque composante $X \in S, T, U, V, A, B$ sont uniques (*i.e.* pointent tous vers le même coup). Pour les trois égalités, les deux termes donnent les suites de coups respectives s_1 , s_2 et s_3 . Si les pointeurs des coups initiaux ne sont pas préservés (ce qui est le cas pour la seconde et la troisième égalité), ceux-ci sont calculés de la même manière avec le Lemme 7.1.9 sur s . \square

7.1.6 Composition verticale

Dans cette sous-section, on va définir la notion de composition verticale entre stratégies et montrer quelque unes de ses propriétés vis-à-vis de la totalité et de l'innocence.

Définition 7.1.11 (Stratégie naturelle). *Soit S, T des arènes ouvertes, et $\eta : S \Rightarrow T$ une stratégie. η est une stratégie naturelle en X (ou juste naturelle si le contexte n'est pas ambigu) si elle est innocente et si pour tout $sab \in \eta$, $a = \square_X$ si et seulement si $b = \square_X$. De plus, η doit répondre à toute occurrence de \square_X .*

Remarque. La définition précédente revient à dire que η est une stratégie fonctorielle par rapport à la décomposition $S \Rightarrow T = (S[X] \Rightarrow T[X])(\square_X)$, et qu'elle est totale sur sa composante argument.

Notation. On a vu que la restriction fonctorielle pouvait fort bien extraire des parties de longueur impaire à partir de parties de longueur paire. On introduit donc la notation :

$$s \in {}^O \eta \Leftrightarrow s \in \eta \cup \text{dom}(\eta)$$

Autrement dit, s est soit une partie de η soit une extension immédiate d'une partie de η . Ainsi, on pourra gérer de manière uniforme les cas où la restriction verticale donne une partie de longueur paire et les cas où elle donne une partie terminant par une unique occurrence de \square_X .

Lemme 7.1.15. *Si $\eta : S \Rightarrow T$ est une stratégie naturelle et si $\sigma : A \Rightarrow B$ est innocente, alors l'ensemble des P -vues P -fonctorielles $s \in \mathcal{L}_{S[B,A] \Rightarrow T[A,B]}$ de longueur paire telles que $s \upharpoonright_{S \Rightarrow T} \in {}^O \eta$ et $s \upharpoonright_{A \Rightarrow B} \in \sigma$ est déterministe, c'est-à-dire que la structure d'arbre correspondante ne branche qu'aux coups d'Opposant.*

Démonstration. Par l'absurde, supposons qu'on ait une P -vue s satisfaisant les contraintes ci-dessus, et telles que deux autres P -vues $sab \neq sab'$ les satisfassent aussi. On distingue les cas :

- Si b est un coup de la composante fonctorielle. Dans ce cas, c'est aussi le cas de b' (sinon b' serait un saut). On a alors $sa \upharpoonright_{S[X] \Rightarrow T[X]} = sa$, et $b = b'$ donne une contradiction immédiate avec l'innocence et le déterminisme de η .
- Si b est un coup de la composante argument, c'est nécessairement aussi le cas de b' (sinon, ce serait un saut). Deux sous-cas sont possibles. Soit on a $b \neq b'$, soit ce sont leurs pointeurs qui diffèrent.

- Si $b \neq b'$, alors $sab \upharpoonright_{A \Rightarrow B} = (sa) \upharpoonright_{A \Rightarrow B} b$ et $(sa) \upharpoonright_{A \Rightarrow B} b'$ sont deux P -vues de σ ne différent que par leur dernier coup Joueur, ce qui est impossible par innocence et déterminisme de σ .
- Si $b = b'$ mais ces deux coups n'ont pas le même pointeur dans sa , il reste deux sous-cas :
 - Si ce sont des coups initiaux de la composante argument, par P -fonctorialité ce sont nécessairement les premiers avec cette polarité dans la composante argument, sinon b ou b' serait un saut. On rappelle que dans la définition de la restriction fonctorielle, on place une occurrence de \square_X à la place de tels coups initiaux, avec le pointeur qu'avait ce coup initial. On examine donc $(sab) \upharpoonright_{S[X] \Rightarrow T[X]}$ et $(sab') \upharpoonright_{S[X] \Rightarrow T[X]}$. Par construction de la restriction, ces deux P -vues sont égales jusqu'aux occurrences finales respectives de \square_X , qui ne diffèrent que par leurs pointeurs ou adresses. Or ces P -vues doivent être toutes deux dans η par hypothèse. On a donc contradiction avec le déterminisme et l'innocence de η .
 - Si ce sont des coups non initiaux, alors par définition de la restriction argument ces pointeurs distincts sont préservés dans le calcul de $(sab) \upharpoonright_{A \Rightarrow B}$ et $(sab') \upharpoonright_{A \Rightarrow B}$, et on a encore contradiction avec l'innocence et le déterminisme de σ .

□

Définition 7.1.12. Soit $\eta : S \Rightarrow T$ une stratégie naturelle, et soit $\sigma : A \Rightarrow B$ innocente. Notons ici $F_{S[A] \Rightarrow T[B]}$ l'ensemble des P -vues bi-fonctorielles sur $S[A] \Rightarrow T[B]$. On définit la composée verticale $\eta[\sigma]$ par :

$$\eta[\sigma] = tr(\{s \in F_{S[A] \Rightarrow T[B]} \mid s \upharpoonright_{S[X] \Rightarrow T[X]} \in^O \eta \wedge s \upharpoonright_{A \Rightarrow B} \in \sigma\})$$

On obtient bien une stratégie $\eta[\sigma] : S[B, A] \Rightarrow T[A, B]$. Par définition, c'est une stratégie saturée. Par le Lemme 7.1.15, c'est une stratégie déterministe.

Notation. Pour la définition ci-dessus, on a fixé implicitement un nom X et on n'a considéré que les trous étiquetés X . Bien sûr, S et T peuvent avoir d'autres trous. Dans les situations plus ambiguës, on notera $\eta[\sigma/X]$ ou $\eta[\sigma/Y]$ pour spécifier les trous vis-à-vis desquels on fera la composition verticale. Il faudra évidemment le cas échéant que η soit naturelle en X et en Y .

Totalité. On a donc une notion de composition verticale qui préserve l'innocence. Il va nous falloir maintenant montrer qu'elle préserve également la totalité : d'une part ce sera utile plus tard pour notre modèle des preuves, mais c'est aussi une bonne façon de valider la définition. En effet, la préservation de la totalité est une bonne façon de vérifier qu'on ne perd pas d'information au cours de la construction. C'est facile de produire des stratégies innocentes arbitraires, mais c'est beaucoup plus dur de produire des stratégies totales, de par leur relation avec les preuves.

Proposition 7.1.16. Si $\eta : S \Rightarrow T$ et $\sigma : A \Rightarrow B$ sont innocentes et totales, alors $\eta[\sigma]$ est totale.

Démonstration. On sait déjà que $\eta[\sigma]$ est innocente. Soit donc s une P -vue de $\eta[\sigma]$, et soit sa une extension valide de s . On va extraire de η et σ une réponse valide de $\eta[\sigma]$. Distinguons les cas :

- Si a est un coup de la composante foncteur, alors sa est entièrement incluse dans la composante foncteur. Elle est donc stable par restriction argument, et est donc une P -vue de η . Par totalité, il existe alors un coup b tel que $sab \in \eta$. Deux sous-cas se présentent :
 - Soit b n'est pas un trou. Dans ce cas, il est immédiat que $sab \in \eta[\sigma]$.
 - Soit $b = \square_X$. Par naturalité de η , on devrait alors avoir $a = \square_X$, ce qui est impossible puisque $sa \in \text{dom}(\eta[\sigma])$, et il n'y a pas de trous dans les parties de $\mathcal{L}_{S[A] \Rightarrow T[B]}$.
- Si a est un coup de la composante argument, alors il apparaît dans $(sa)_{\uparrow A \Rightarrow B} \in \sigma$. Comme σ est totale, il existe b tel que $sab \in \sigma$. Encore une fois, deux sous-cas se présentent.
 - b est non-initial. Dans ce cas on le place dans sa avec le même pointeur, et $sab \in \eta[\sigma]$ par construction.
 - b est initial (dans ce cas, c'est forcément un coup initial de A). Distinguons encore deux cas :
 - b n'est pas le premier coup initial de A dans s . Dans ce cas on trouve dans sa d'autres coups initiaux de A , et ils pointent tous au même endroit par le Lemme 7.1.4. On étend donc sa en sab en faisant pointer b à ce même endroit, et par définition de la restriction et de la composition verticale, on a bien $sab \in \eta[\sigma]$.
 - b est le premier coup initial de A dans s (autrement dit, b ouvre A). On a donc nécessairement $sa_{\uparrow S[X] \Rightarrow T[X]} = s' \square_X^O$. Comme η est naturelle, il existe une occurrence de \square_X^P telle que $s' \square_X^O \square_X^P \in \eta$, sachant que par définition de la restriction fonctorielle, tous les coups de s' sont des coups de la composante fonctorielle déjà présents dans s , avec les mêmes pointeurs. On étend donc sa en sab , où b récupère le pointeur de \square_X^P dans $s' \square_X^O$. Par construction, on aura alors bien $sab \in \eta[\sigma]$. □

Remarque. La preuve ci-dessus revient à donner une réponse à la question suivante : dans une partie $s \in \text{dom}(\eta[\sigma])$, à quelle stratégie est-ce de jouer ? η ou σ ? Dans certains cas il existe une réponse non ambiguë. Par exemple si le dernier coup est dans la composante foncteur, c'est à η de jouer. Si le dernier coup est dans la composante argument, c'est plus compliqué. On donne d'abord à σ l'état actuel de la partie sur la composante argument. Si elle veut jouer un coup non-initial, il existe une unique façon, non ambiguë, de l'ajouter à s . Si σ veut jouer un coup initial, le problème est qu'on ne sait pas où le faire pointer ni quelle adresse lui donner. En fait σ fournit le coup, mais η fournit le pointeur et l'adresse : c'est ceux avec lesquels elle placerait l'occurrence de \square_X correspondante.

Bon parenthésage. La composition verticale produit des stratégies bien parenthésées :

Lemme 7.1.17. *Si $\eta : S \Rightarrow T$ et $\sigma : A \Rightarrow B$ sont bien parenthésées, alors $\eta[\sigma] : S[A] \Rightarrow T[B]$ est bien parenthésée.*

Démonstration. Par construction, les P -vues de $\eta[\sigma]$ sont obtenues par concaténation de P -vues de η et de σ donc de P -vues bien parenthésées, donc elles doivent toujours être bien parenthésées. □

Compositions parallèles. Si S et T sont des arènes ouvertes avec des trous d'étiquettes X et Y et si $\eta : S \Rightarrow T$ est naturelle en ces deux structures fonctorielles, alors on va montrer que les opérations de composition verticale correspondantes sont indépendantes, et commutent.

Lemme 7.1.18 (Compositions parallèles). *Soient S, T des arènes ouvertes avec des occurrences de \square_X , toutes de la même polarité, et des occurrences de \square_Y , également toutes de la même polarité. Soit également $\eta : S \Rightarrow T$ une stratégie naturelle en X et en Y , et soient $\sigma : A \Rightarrow B$ et $\tau : C \Rightarrow D$. Alors, les propriétés suivantes sont vérifiées :*

- $\eta[\sigma/X]$ est naturelle en Y ;
- $\eta[\tau/Y]$ est naturelle en X ;
- $\eta[\sigma/X][\tau/Y] = \eta[\tau/Y][\sigma/X]$.

Démonstration. Trois propriétés à prouver :

- Soit $sab \in \eta[\sigma/X]$ une P -vue de $\eta[\sigma/X]$. Supposons $a = \square_Y$ et examinons alors $sab \upharpoonright_{S[A/X][Y] \Rightarrow T[A/X][Y]}$. C'est nécessairement une P -vue par définition de la restriction fonctorielle, et elle finit également par ab , car a et b sont des coups de la composante fonctorielle pour $S[A/X][Y] \Rightarrow T[A/X][Y]$. On obtient donc une P -vue $s'ab \in \eta$ par définition de la composition verticale. Mais η est naturelle en Y , donc on a également $b = \square_Y$. Le même raisonnement montre que si $b = \square_Y$, alors $a = \square_X$: $\eta[\sigma/X]$ est donc naturelle en Y .
- Même raisonnement qu'au dessus.
- Les deux inclusions sont symétriques, on va donc juste prouver l'inclusion suivante :

$$\eta[\sigma/X][\tau/Y] \subseteq \eta[\tau/Y][\sigma/X]$$

Ces deux stratégies étant innocentes, on va se contenter de le montrer sur leurs P -vues. Soit donc $s \in \eta[\sigma/X][\tau/Y]$ une P -vue. Deux choses à montrer :

- Montrons que $s \upharpoonright_{S[D/Y^-, C/Y^+][X] \Rightarrow T[C/Y^-, D/Y^+][X]} \in^O \eta[\tau/Y]$. On calcule en utilisant les égalités du Lemme 7.1.13 :

$$\begin{aligned} (s \upharpoonright_{S[D/Y^-, C/Y^+][X] \Rightarrow T[C/Y^-, D/Y^+][X]}) \upharpoonright_{S[Y] \Rightarrow T[Y]} &= \\ (s \upharpoonright_{S[B/X^-, A/X^+][Y] \Rightarrow T[A/X^-, B/X^+][Y]}) \upharpoonright_{S[X] \Rightarrow T[X]} &\in^O \eta \end{aligned}$$

$$\begin{aligned} (s \upharpoonright_{S[D/Y^-, C/Y^+][X] \Rightarrow T[C/Y^-, D/Y^+][X]}) \upharpoonright_{B \Rightarrow C} &= s \upharpoonright_{B \Rightarrow C} \\ &\in \tau \end{aligned}$$

- Montrons maintenant que $s \upharpoonright_{A \Rightarrow B} \in \sigma$. C'est directement une égalité du Lemme 7.1.13.

$$\begin{aligned} s \upharpoonright_{A \Rightarrow B} &= (s \upharpoonright_{S[D/Y^-, C/Y^+][X] \Rightarrow T[C/Y^-, D/Y^+][X]}) \upharpoonright_{A \Rightarrow B} \\ &\in \sigma \end{aligned}$$

□

Compositions en cascade. Tout comme on pouvait effectuer des substitutions d'arènes successives, on peut effectuer des compositions verticales successives. La propriété suivante énonce en quelque sorte l'*associativité* de la composition verticale :

Lemme 7.1.19 (Composition verticales en cascade). *Soient S, T, U, V des arènes ouvertes à variance fixe, où S, T ont même polarité et U, V ont même polarité. Soient également $\eta : S \Rightarrow T$ et $\epsilon : U \Rightarrow V$ deux stratégies naturelles et $\sigma : A \Rightarrow B$ une stratégie innocente. Alors :*

$$(\eta[\epsilon])[\sigma] = \eta[\epsilon[\sigma]]$$

Démonstration. Pour que le terme de gauche ait du sens, il faut d'abord s'assurer que $\eta[\epsilon]$ est bien une stratégie naturelle, mais c'est une conséquence directe du fait que ϵ est naturelle. Les égalités à montrer sont des applications directes du Lemme 7.1.14. \square

7.1.7 Décomposition des stratégies fonctorielles

Soient S, T des foncteurs ouverts, et $\sigma : S[B, A] \Rightarrow T[A, B]$ une stratégie fonctorielle. L'objectif de cette sous-section est d'examiner dans quels cas l'opération de composition verticale est *inversible*, c'est-à-dire dans quels cas on peut décomposer σ en deux stratégies $\sigma_f : S \Rightarrow T$ et $\sigma_a : A \Rightarrow B$ telle que $\sigma = \sigma_f[\sigma_a]$. Cet examen se révélera utile plus tard lorsqu'on voudra prouver le théorème de factorisation des transformations naturelles.

Par définition, chacune des parties O -fonctorielles de σ est en fait bi-fonctorielle, on peut donc en considérer les restrictions verticales. De même, le Lemme 7.1.4 montre que les P -vues de σ seront nécessairement de la forme $s_1 s_2$, avec s_1 dans la composante foncteur et s_2 dans la composante argument : σ générera alors un ensemble de P -vues sur $S \Rightarrow T$ ainsi qu'un ensemble de P -vues sur $A \Rightarrow B$. Toutefois, il est loin d'être garanti que ces ensembles de vues correspondent à des stratégies innocentes. On va donner maintenant une notion d'*uniformité* sur σ , caractérisant son comportement par rapport à de telles restrictions.

Définition 7.1.13. Soit $\sigma : S[B, A] \Rightarrow T[A, B]$ une stratégie fonctorielle. On définit la restriction fonctorielle de σ par :

$$\sigma_{\upharpoonright S \Rightarrow T} = \text{tr}(\{s_{\upharpoonright S \Rightarrow T} \mid s \in \ulcorner \sigma \urcorner\})$$

Lemme 7.1.20. Soit $\sigma : S[B, A] \Rightarrow T[A, B]$ fonctorielle, alors $\sigma_{\upharpoonright S \Rightarrow T}$ est innocente.

Démonstration. Par définition elle est saturée, il reste juste à montrer qu'elle est déterministe. Si le dernier coup n'est pas un trou, la partie à jouer est une partie de σ et l'unicité du choix découle immédiatement du déterminisme de σ . Sinon le dernier coup est un trou \square_X , mais si $\sigma_{\upharpoonright S \Rightarrow T}$ lui répond ça doit être également un trou, sinon ce coup proviendrait d'une P -vue non bi-fonctorielle de σ . De plus le choix de l'adresse et du pointeur sont uniques car deux choix possibles proviendraient de deux P -vues s_1 et s_2 de σ coïncidant sur $S \Rightarrow T$, et ouvrant des copies différentes de A à partir de la même copie de B . En combinant ces deux P -vues, il serait alors facile de construire une partie O -fonctorielle de σ avec un saut de type 4. \square

Remarque. Dans l'optique d'inverser l'opération de composition verticale, il faut se demander si $\sigma_{\upharpoonright S \Rightarrow T}$ est naturelle, c'est-à-dire si elle répond immédiatement à toute occurrence de \square_X par une autre occurrence de \square_X . Malheureusement, la réponse est non : si σ ne joue jamais sur A , il est impossible de savoir quels seraient le pointeur et l'adresse de l'éventuelle seconde occurrence de \square_X dans $\sigma_{S \Rightarrow T}$. La Figure 7.6 décrit un exemple de stratégie σ dont la restriction fonctorielle n'est pas naturelle, puisqu'elle n'a rien à répondre à \square_X . Cet exemple est quelque peu mal choisi puisqu'ici il y a une unique façon possible d'étendre $\sigma_{\upharpoonright S \Rightarrow T}$ en une stratégie naturelle, cependant il suffirait de rajouter une seconde copie de **bool** à gauche pour que ce ne soit plus le cas.

Il existe cependant une classe notable de stratégies $\sigma : S[B, A] \Rightarrow T[A, B]$ où il est clair que $\sigma_{\upharpoonright S \Rightarrow T}$ est naturelle : il s'agit des stratégies *strictes sur* $A \Rightarrow B$, c'est-à-dire que σ répond immédiatement à tout coup initial de B par un coup initial de A . Ce n'est pas la classe de stratégie la plus large assurant cette naturalité : il suffirait en effet d'assurer que A soit

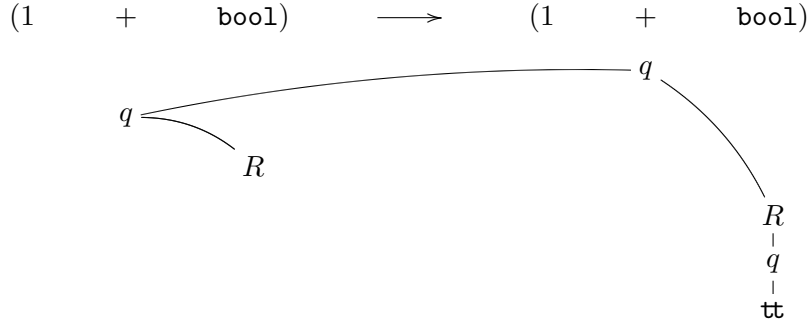


FIG. 7.6 – Stratégie fonctorielle sur $(1 + X)[\mathbf{bool}] \Rightarrow (1 + X)[\mathbf{bool}]$, dont la restriction fonctorielle n'est pas naturelle.

forcément visité un jour, apportant ainsi l'information nécessaire pour compléter $\sigma_{\upharpoonright S \Rightarrow T}$. Mais cette généralité supplémentaire viendrait avec son lot de complications techniques, aussi on va l'éviter ici.

Lemme 7.1.21. *Supposons que A et B soient des arènes non vides². Soit $\sigma : S[B, A] \Rightarrow T[A, B]$ une stratégie fonctorielle, et stricte sur $A \Rightarrow B$, alors $\sigma_{\upharpoonright S \Rightarrow T}$ est une stratégie naturelle.*

Démonstration. Par définition de la restriction verticale, toutes les parties de longueur paire produites respectent la contrainte de naturalité. Le fait que $\sigma_{\upharpoonright S \Rightarrow T}$ répond nécessairement aux occurrences de \square_X vient de fait que σ est stricte, et la localisation du coup initial (qui existe, puisque A est non vide) sur A donne la localisation de l'occurrence de \square_X à jouer. \square

Uniformité argument. Prenons $\sigma : S[B, A] \Rightarrow T[A, B]$ fonctorielle, sa restriction $\sigma_f : S \Rightarrow T$ est donc innocente. On veut maintenant examiner les cas où on peut récupérer $\sigma_a : A \Rightarrow B$, c'est-à-dire les cas où la dynamique de σ sur la composante argument ne dépend pas de la composante fonctorielle. Pour commencer, il est très simple de récupérer à partir de σ des «sous-stratégies» innocentes sur $A \Rightarrow B$: il suffit de fixer l'histoire sur la composante fonctorielle, ainsi on sera sûr de récupérer une stratégie innocente sur la composante argument. C'est la *décomposition par tranches*.

Définition 7.1.14 (Tranche). *Soit $\sigma : S[B, A] \Rightarrow T[A, B]$ une stratégie fonctorielle. Une tranche de σ est une P -vue bi-fonctorielle $t \in \ulcorner \sigma_{\upharpoonright S \Rightarrow T} \urcorner$ maximale, c'est-à-dire qu'elle finit par deux occurrences de \square_X .*

Définition 7.1.15 (Restriction par tranches). *Soit $\sigma : S[B, A] \Rightarrow T[A, B]$ une stratégie fonctorielle, et t une tranche de σ . La restriction argument de σ sur la tranche t est définie par :*

$$\sigma_{\upharpoonright^t A \Rightarrow B} = \text{tr}(\{s_{\upharpoonright A \Rightarrow B} \mid s \in \ulcorner \sigma \urcorner \wedge \ulcorner s_{\upharpoonright S \Rightarrow T} \urcorner = t\})$$

Lemme 7.1.22. *Soit $\sigma : S[B, A] \Rightarrow T[A, B]$ une stratégie fonctorielle et t une tranche de σ , alors $\sigma_{\upharpoonright^t A \Rightarrow B}$ est une stratégie innocente.*

²Ici et plus tard, on dira qu'une arène est *non vide* si elle admet au moins un coup initial.

Démonstration. Par définition, $\sigma \upharpoonright_{A \Rightarrow B}$ est saturée. Il reste donc à vérifier qu'elle est déterministe. Étant donné que t étant une tranche, elle est de la forme $t' \square_X \square_X$. Alors, chaque P -vue s de $\sigma \upharpoonright_{A \Rightarrow B}$ correspond à une P -vue $\lceil t' \rceil s$ de σ où on a fait pointer les coups initiaux de B et A vers (respectivement) la première puis la seconde occurrence de \square_X dans t . Le déterminisme de l'ensemble de ces P -vues correspond donc directement au déterminisme de l'ensemble des P -vues de σ commençant par t' . \square

Définition 7.1.16. *On dit qu'une stratégie $\sigma : S[B, A] \Rightarrow T[A, B]$ est uniforme si elle est fonctorielle et si pour toutes tranches t_1 et t_2 de σ , on a $\sigma \upharpoonright_{t_1 S \Rightarrow T} = \sigma \upharpoonright_{t_2 S \Rightarrow T}$.*

Dans le cas des stratégies uniformes, on a donc une façon canonique de récupérer une stratégie sur $S \Rightarrow T$ et une autre sur $A \Rightarrow B$. Cette opération est *presque* l'inverse de la composition verticale, notons juste le cas dégénéré représenté en Figure 7.6, où la composition $\eta[\sigma]$ perd de l'information sur η ($(\eta[\sigma]) \upharpoonright_{S \Rightarrow T}$ n'est plus totale) parce que σ n'atteint jamais A . Comme dit plus haut, on se contentera de le montrer dans le cas des stratégies *strictes sur* $A \Rightarrow B$.

Proposition 7.1.23. *Supposons que les arènes A et B soient non vides. Si $\sigma : S[B, A] \Rightarrow T[A, B]$ est uniforme et stricte sur $A \Rightarrow B$, l'égalité suivante est vérifiée :*

$$\sigma \upharpoonright_{S \Rightarrow T} [\sigma \upharpoonright_{A \Rightarrow B}] = \sigma$$

De même, si $\eta : S \Rightarrow T$ est une stratégie naturelle et $\sigma : A \Rightarrow B$ une stratégie innocente stricte³,

$$\begin{aligned} \eta[\sigma] \upharpoonright_{S \Rightarrow T} &= \eta \\ \eta[\sigma] \upharpoonright_{A \Rightarrow B} &= \sigma \end{aligned}$$

On a donc bien un isomorphisme entre l'ensemble des stratégies uniformes strictes sur $A \Rightarrow B$ et des paires (η, σ) où η est une stratégie naturelle et σ est une stratégie innocente stricte.

Démonstration. La première égalité est triviale. Pour les deux suivantes, notons que l'inclusion $\eta \subseteq \eta[\sigma] \upharpoonright_{S \Rightarrow T}$ est vraie uniquement parce que σ atteint A , sans quoi aucune partie de $\eta[\sigma]$ ne permettrait de savoir où η place le pointeur de son occurrence de \square_X . \square

7.2 Foncteurs ouverts et stratégies naturelles

Pour le moment, on a défini une notion de composition verticale essentiellement statique. Certes, elle permet de définir de nouvelles stratégies innocentes, totales et bien parenthésées, mais ce n'est pas d'un grand intérêt si on ne sait pas comment ces nouvelles stratégies se comportent vis-à-vis de la composition. C'est l'objet de cette section : on va voir que les stratégies fonctorielles obtenues par composition verticale vérifient une «loi d'échange» très similaire à la loi d'échange entre les deux types de composition, dans un cadre 2-catégorique. La terminologie est un peu maladroite car notre composition verticale entre stratégies correspond à la composition *horizontale* des 2-cellules, mais ce désaccord paraît cependant incontournable : il paraîtrait incongru de qualifier de «composition verticale» la composition usuelle entre stratégies.

³C'est-à-dire qu'elle répond immédiatement au coup initial sur B par un coup initial sur A .

Comme expliqué dans l'introduction de ce chapitre, cette loi d'échange permettra de définir d'un coup les foncteurs ouverts, ainsi qu'une classe de transformations naturelles entre ces foncteurs. On donnera alors quelques exemples de foncteurs ouverts : en particulier, on verra que tous les foncteurs usuels obtenus par \times , $+$ ou \Rightarrow sont ouverts. Finalement, on montrera un puissant *théorème de factorisation* : toutes les transformations naturelles entre foncteurs ouverts sont en fait des «transformations naturelles ouvertes», c'est-à-dire que leurs composantes sont obtenues par composition verticale.

7.2.1 Loi d'échange

Lemme 7.2.1 (Loi d'échange – cas positif). *Soient S, T, U des arènes ouvertes telles que \square_X n'apparaît qu'en polarité Opposant dans S, T et U . Soient également $\eta : S \Rightarrow T$ et $\epsilon : T \Rightarrow U$ des stratégies naturelles et $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ des stratégies innocentes. Alors :*

$$(\eta; \epsilon)[\sigma; \tau] = \eta[\sigma]; \epsilon[\tau]$$

Démonstration. Par le Lemme 7.1.5, il suffit de montrer les deux inclusions pour les parties bi-fonctorielles. Montrons que $(\eta; \epsilon)[\sigma; \tau] \subseteq \eta[\sigma]; \epsilon[\tau]$. Soit $s \in (\eta; \epsilon)[\sigma; \tau]$ bi-fonctorielle. Par définition de la composition verticale, on a $s_{\uparrow S[X] \Rightarrow U[X]} \in^O \eta; \epsilon$. Soit u_1 le témoin pour cette interaction⁴. De même, on a $s_{\uparrow A \Rightarrow C} \in \sigma; \tau$, soit u_2 le témoin de cette interaction. On distingue alors quatre cas, suivant que u_1 atteint ou pas les trous présents. Par la condition de polarité des trous de S, T et U et par functorialité, u_1 peut soit ne pas atteindre de trous, soit atteindre dans l'ordre celui de U , puis T , puis S . Les cas possibles sont donc les suivants :

- Si u_2 est vide, alors nécessairement u_1 ne contient pas d'occurrence des trous. Donc u_1 est déjà un témoin valide pour s dans $\eta[\sigma] \parallel \epsilon[\tau]$.
- Si u_2 reste dans C , alors nécessairement $u_1 = u'_1 \square_X$ finit par une occurrence du trou dans U . Dans ce cas on définit $u = u'_1 u_2$, et les coups initiaux dans u_2 pointent comme pointait l'occurrence de \square_X dans u_1 . On vérifie immédiatement que $u \in \eta[\sigma] \parallel \epsilon[\tau]$.
- Si u_2 passe dans C et B (mais pas dans A), on a encore nécessairement $u_1 = u'_1 \square_X$, avec u'_1 sans trous. Cependant, il existe une unique façon d'étendre u_1 en $u'_1 \square_X^{(U)} \square_X^{(T)}$ de sorte que $(u'_1 \square_X^{(U)} \square_X^{(T)})_{\uparrow T, U} \in \epsilon$. On pose $u = u'_1 u_2$, où les coups initiaux de u_2 dans B et C pointent respectivement comme $\square_X^{(T)}$ et $\square_X^{(U)}$ dans la partie ci-dessus. Encore une fois, une vérification immédiate montre qu'on a bien $u \in \eta[\sigma] \parallel \epsilon[\tau]$.
- Dernier cas : u_2 visite A, B et C . Dans ce cas nécessairement u_1 est de la forme $u'_1 \square_X^{(U)} \square_X^{(T)} \square_X^{(S)}$. On pose cette fois encore $u = u'_1 u_2$ où les pointeurs des coups initiaux dans A, B et C de u_2 pointent respectivement comme $\square_X^{(S)}$, $\square_X^{(T)}$ et $\square_X^{(U)}$ dans la partie ci-dessus. On a bien $u \in \eta[\sigma] \parallel \epsilon[\tau]$.

Reste à montrer l'autre inclusion. Soit $s \in \eta[\sigma]; \epsilon[\tau]$ bi-fonctorielle. Il existe un unique témoin $u \in \eta[\sigma] \parallel \epsilon[\tau]$ pour s , qu'on peut décomposer en une partie fonctorielle $Fonc(u)$ et une partie argument $Arg(u)$. Mais alors, les équations des Lemmes 7.1.8 et 7.1.11 assurent directement que $Fonc(u)$ est un témoin pour $s_{\uparrow S[X] \Rightarrow U[X]}$ dans $\eta \parallel \epsilon$ et que $Arg(u)$ est un témoin pour $s_{\uparrow A \Rightarrow C}$ dans $\sigma \parallel \tau$, ce qui conclut la preuve. \square

⁴Si $s_{\uparrow S[X] \Rightarrow U[X]}$ est de longueur impaire, u_1 est le témoin dans $\eta \parallel \epsilon$ de son préfixe immédiat, étendu par le dernier coup de s .

Lemme 7.2.2 (Loi d'échange – cas négatif). *Soient S, T, U des arènes ouvertes telles que \square_X n'apparaît qu'en polarité Joueur dans S, T et U . Soient également $\eta : S \Rightarrow T$ et $\epsilon : T \Rightarrow U$ des stratégies naturelles et $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ des stratégies innocentes. Alors :*

$$(\eta; \epsilon)[\sigma; \tau] = \eta[\tau]; \epsilon[\sigma]$$

Démonstration. Preuve similaire au cas positif. □

Lemme 7.2.3. *Soit T une arène ouverte, alors id_T est une stratégie naturelle.*

Démonstration. Trivial. □

Proposition 7.2.4 (Préservation de l'identité). *Soit T une arène ouverte de variance fixe, alors pour toute arène A , $id_T[id_A] = id_{T[A/X]}$.*

Démonstration. Ce sont deux stratégies totales, il suffit donc par le Lemme 2.1.1 de montrer une inclusion pour avoir l'égalité. Prenons donc une P -vue s de $id_T[id_A]$. Par le Lemme 7.1.4, elle est de la forme $s_1 s_2$, où s_1 est une P -vue de id_T et s_2 une P -vue de id_A (aux pointeurs près). Or, les pointeurs des coups initiaux de A doivent pointer sur le même coup à gauche et à droite, par définition de id_T et de la composition verticale. Donc on doit avoir $s_{\uparrow L} = s_{\uparrow R}$, et on a bien $s \in id_{T[A/X]}$. L'égalité s'ensuit. □

Théorème 7.2.5 (Foncteurs ouverts). *Soit T une arène ouverte de polarité positive, c'est à dire que toutes les occurrences de \square_X apparaissent en polarité Opposant. On pose :*

- Si A est une arène, $T(A) = T[A]$;
- Si $\sigma : A \Rightarrow B$ est une stratégie, $T(\sigma) = id_T[\sigma] : T[A] \Rightarrow T[B]$.

Alors l'action de T sur les stratégies est fonctorielle. On appelle foncteurs ouverts les foncteurs ainsi définis par composition verticale. On définit de la même façon les foncteurs ouverts contravariants.

Démonstration. On prouve le cas où T est covariant, l'autre cas est identique. Soient $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies innocentes. On a :

$$\begin{aligned} T(\sigma; \tau) &= id_T[\sigma; \tau] \\ &= (id_T; id_T)[\sigma; \tau] \\ &= id_T[\sigma]; id_T[\tau] \\ &= T(\sigma); T(\tau) \end{aligned}$$

La préservation des identités est assurée par la Proposition 7.2.4. □

Théorème 7.2.6 (Transformations naturelles ouvertes). *Soient S, T des arènes ouvertes à variance fixe. Soit $\eta : S \Rightarrow T$ une stratégie naturelle. La famille :*

$$(\eta[id_A])_A : S(A) \Rightarrow T(A)$$

est naturelle en A . On appelle transformations naturelles ouvertes les transformations naturelles entre foncteurs ouverts ainsi définies par l'action d'une stratégie naturelle.

Démonstration. Encore une fois, on fait la preuve dans le cas où T est covariant, l'autre cas est identique. Si $\sigma : A \Rightarrow B$, on calcule :

$$\begin{aligned}
 \eta_A; T(\sigma) &= \eta[id_A]; id_T[\sigma] \\
 &= (\eta; id_T)[id_A; \sigma] \\
 &= \eta[\sigma] \\
 &= (id_S; \eta)[\sigma; id_B] \\
 &= id_S[\sigma]; \eta[id_B] \\
 &= S(\sigma); \eta_B
 \end{aligned}$$

□

Parenthésage. Pour le moment, on sait associer à tout arène ouverte T de variance fixe un foncteur $T : \mathbf{Inn} \rightarrow \mathbf{Inn}$ si les trous dans T sont de polarité Opposant, et $T : \mathbf{Inn}^{op} \rightarrow \mathbf{Inn}$ si les trous dans T sont de polarité Joueur. Notons tout d'abord qu'on peut à volonté remplacer \mathbf{Inn} par \mathbf{Par} ici, puisqu'on a vu que l'opération de composition verticale préservait également le bon parenthésage.

Variance mixte. Soit T une arène ouverte avec une seule variable X , mais apparaissant dans des polarités différentes. On peut alors considérer T comme une arène avec deux variables libres X^- et X^+ , mais à variance fixe en chacune de ces variables. Le Lemme 7.1.18 établit alors que T se comporte comme un bifoncteur en ces deux variables, ainsi on définit $T : \mathbf{Par}^{op} \times \mathbf{Par} \rightarrow \mathbf{Par}$, comme attendu.

Arité multiple. Le même Lemme 7.1.18 garantit maintenant que si T est une arène ouverte avec des trous étiquetés X_1, \dots, X_n , on en déduit un foncteur :

$$T : (\mathbf{Par}^{op} \times \mathbf{Par})^n \rightarrow \mathbf{Par}$$

Ce qui conclut la construction des foncteurs ouverts.

7.2.2 Restriction et composition

Dans les développements techniques qui suivent, il sera fondamental de pouvoir étudier des propriétés dynamiques sur les différentes composantes de stratégies fonctorielles, mais non obtenues par composition verticale. Plus concrètement, imaginons qu'on dispose d'un diagramme commutatif :

$$\begin{array}{ccc}
 S[A] & \xrightarrow{\sigma} & T[B] \\
 \delta \downarrow & & \downarrow \tau \\
 V[D] & \xrightarrow{\gamma} & U[C]
 \end{array}$$

où les stratégies σ, δ, τ et γ sont fonctorielles, c'est-à-dire qu'on peut appliquer le mécanisme de restriction introduit en Section 7.1.7. On aimerait alors pouvoir déduire de la commutativité de ce diagramme des informations sur le comportement de ces stratégies sur les différentes composantes. Pour cela, il est nécessaire d'étudier la décomposition des stratégies fonctorielles de la Section 7.1.7 en conjonction avec la composition des stratégies.

Lemme 7.2.7. *Supposons que les arènes A, B et C soient non vides. Soient $\sigma : S[A] \Rightarrow T[B]$ et $\tau : T[B] \Rightarrow U[C]$ deux stratégies fonctorielles, respectivement strictes sur $A \Rightarrow B$ et $B \Rightarrow C$. Alors :*

$$(\sigma; \tau)_{\upharpoonright S \Rightarrow U} = \sigma_{\upharpoonright S \Rightarrow T}; \tau_{\upharpoonright T \Rightarrow U}$$

Démonstration. Pour la preuve, on va supposer que S, T et U sont covariants : la démonstration est similaire pour le cas contravariant. On sait déjà que puisque σ et τ sont des stratégies fonctorielles, c'est aussi le cas de $\sigma; \tau$. Montrons que les deux stratégies considérées ont les mêmes P -vues. Soit donc $s \in \ulcorner (\sigma; \tau)_{\upharpoonright S \Rightarrow U} \urcorner$. Le style de preuve est maintenant habituel : par définition, il existe alors $s' \in \ulcorner \sigma; \tau \urcorner$ tel que $s' = s_{\upharpoonright S \Rightarrow U}$. Soit alors u le témoin pour s' dans $\sigma \parallel \tau$, u est alors bi-fonctorielle donc est de la forme $u_1 u_2$, où les coups initiaux de u_2 de $X \in \{A, B, C\}$ partagent le même pointeur et le même préfixe d'adresse. On en déduit $u_1 \square_X \square_X \square_X$, où les pointeurs (et les adresses) des occurrences de \square_X sont définis par ceux des coups initiaux sur C, B, A . Par construction, c'est alors un témoin pour s dans $\sigma_{\upharpoonright S \Rightarrow T} \parallel \tau_{\upharpoonright T \Rightarrow U}$. Réciproquement, soit $s \in \ulcorner \sigma_{\upharpoonright S \Rightarrow T}; \tau_{\upharpoonright T \Rightarrow U} \urcorner$. Pour la même raison qu'au dessus, on s'intéresse uniquement au cas où $s = s' \square_X \square_X$. Par functorialité de σ et τ , son témoin u a alors la même forme $u' \square_X \square_X \square_X$. Utilisons maintenant le fait qu'il existe un coup initial i_C sur C , par conséquent on peut substituer la première de ces trois occurrences de \square_X par celui-ci. Mais σ et τ sont strictes respectivement sur $A \Rightarrow B$ et $B \Rightarrow C$, et ont ainsi une réaction i_B à i_C puis i_A à i_B . On peut alors vérifier que $u' i_C i_B i_A$ est dans $\sigma \parallel \tau$, et que c'est un témoin pour $s' i_C i_A$, or $(s' i_C i_A)_{\upharpoonright S \Rightarrow T} = s$. \square

Lemme 7.2.8. *Soient $\sigma : S[A] \Rightarrow T[B]$ et $\tau : T[B] \Rightarrow U[C]$ deux stratégies fonctorielles, alors il existe une injection qui à chaque tranche t de $\sigma; \tau$ associe une tranche $\phi_1(t)$ de σ et une tranche $\phi_2(t)$ de τ .*

Démonstration. Par définition, t est une P -vue maximale de $(\sigma; \tau)_{\upharpoonright S \Rightarrow U}$. Par le Lemme 2.1.6, t a un témoin unique $u \in \sigma \parallel \tau$. On pose alors $\phi_1(t) = \ulcorner u_{\upharpoonright S[A] \Rightarrow T[B]} \urcorner$ et $\phi_2(t) = \ulcorner u_{\upharpoonright T[B] \Rightarrow U[C]} \urcorner$. Par construction, elles sont bien maximales et ce sont bien des tranches respectives de σ et τ . \square

La séparation d'une tranche en deux donnée par le lemme précédent n'est pas choisie au hasard, c'est celle qui valide le lemme suivant :

Lemme 7.2.9. *Soient $\sigma : S[A] \Rightarrow T[B]$ et $\tau : T[B] \Rightarrow U[C]$ deux stratégies fonctorielles et soit t une tranche de $\sigma; \tau$. Si S, T, U sont covariants, on a l'égalité suivante :*

$$(\sigma; \tau)_{\upharpoonright^t A \Rightarrow C} = \sigma_{\upharpoonright^{\phi_1(t)} A \Rightarrow B}; \tau_{\upharpoonright^{\phi_2(t)} B \Rightarrow C}$$

S'ils sont contravariants, on a l'égalité duale :

$$(\sigma; \tau)_{\upharpoonright^t C \Rightarrow A} = \tau_{\upharpoonright^{\phi_2(t)} C \Rightarrow B}; \sigma_{\upharpoonright^{\phi_1(t)} B \Rightarrow A}$$

S'ensuit immédiatement la stabilité par composition des stratégies uniformes.

Démonstration. Détaillons la preuve dans le cas covariant. Soit $s \in \ulcorner (\sigma; \tau)_{\upharpoonright^t A \Rightarrow C} \urcorner$. Par définition de la restriction et par propriété des P -vues fonctorielles, on a donc $s' \in \ulcorner \sigma; \tau \urcorner$ telle que $\ulcorner s'_{\upharpoonright S \Rightarrow U} \urcorner = t$ et $s'_{\upharpoonright A \Rightarrow C} = s$. Considérons u le témoin unique de s' dans $\sigma \parallel \tau$. Ce témoin est fonctoriel, il se décompose donc en $u_1 u_2$ de la façon habituelle. Soit u'_2 l'interaction

sur $I(A, B, C)$ obtenue en réarrangeant les pointeurs des coups initiaux comme décrit par le Lemme 7.1.9, et u'_1 obtenue en rajoutant les occurrences de \square_X correspondant aux coups initiaux sur C, B, A . Il est alors immédiat que u'_1 est un témoin pour la tranche t , donc par unicité du témoin on a $\ulcorner(u_1)\urcorner_{S \Rightarrow T} = \phi_1(t)$ et $\ulcorner(u_1)\urcorner_{T \Rightarrow U} = \phi_2(t)$, il est alors immédiat que $(u_2)\urcorner_{A \Rightarrow B} \in \sigma_{\ulcorner\phi_1(t)\urcorner_{A \Rightarrow B}}$ et $(u_2)\urcorner_{B \Rightarrow C} \in \tau_{\ulcorner\phi_2(t)\urcorner_{B \Rightarrow C}}$. Donc, u_2 est comme requis un témoin pour s dans $\sigma_{\ulcorner\phi_1(t)\urcorner_{A \Rightarrow B}}; \tau_{\ulcorner\phi_2(t)\urcorner_{B \Rightarrow C}}$. Réciproquement, soit $s \in \sigma_{\ulcorner\phi_1(t)\urcorner_{A \Rightarrow B}}; \tau_{\ulcorner\phi_2(t)\urcorner_{B \Rightarrow C}}$ bi-fonctorielle, prenons u le témoin correspondant. Rappelons alors que $\phi_1(t)$ et $\phi_2(t)$ proviennent du choix d'un témoin $v \square_X \square_X \square_X$ pour la tranche t , on considère alors vu (où les pointeurs des coups initiaux dans u ont été préalablement réimplantés dans v) : il est direct de vérifier que $vu \in \sigma \parallel \tau$. De plus, c'est par construction un témoin pour ts (où les pointeurs des coups initiaux de s ont été réimplantés dans t), donc s est bien une partie de $(\sigma; \tau)\urcorner_{A \Rightarrow C}$. \square

Décomposition de diagrammes commutatifs. Revenons maintenant le diagramme évoqué au début de cette section :

$$\begin{array}{ccc} S[A] & \xrightarrow{\sigma} & T[B] \\ \delta \downarrow & & \downarrow \tau \\ V[D] & \xrightarrow{\gamma} & U[C] \end{array}$$

où les stratégies $\sigma, \tau, \delta, \gamma$ sont supposées fonctorielles, strictes sur leur composante argument, mais pas nécessairement uniformes. On va décomposer ce diagramme en deux diagrammes : un sur la composante fonctorielle, l'autre sur la composante argument. D'une part, le Lemme 7.2.7 nous donne le diagramme commutatif suivant :

$$\begin{array}{ccc} S & \xrightarrow{\sigma_{\ulcorner S \Rightarrow T \urcorner}} & T \\ \delta_{\ulcorner S \Rightarrow V \urcorner} \downarrow & & \downarrow \tau_{\ulcorner T \Rightarrow U \urcorner} \\ V & \xrightarrow{\gamma_{\ulcorner V \Rightarrow U \urcorner}} & U \end{array}$$

D'autre part, les stratégies $\sigma; \tau$ et $\delta; \gamma$ sont égales donc partagent les mêmes tranches : on parle alors de *tranches du diagramme*. Sur chacune de ces tranches, on peut appliquer le Lemme 7.2.9 et obtenir :

$$\begin{array}{ccc} A & \xrightarrow{\sigma_{\ulcorner \phi_1(t) \urcorner_{A \Rightarrow B}}} & B \\ \delta_{\ulcorner \phi'_1(t) \urcorner_{A \Rightarrow D}} \downarrow & & \downarrow \tau_{\ulcorner \phi_2(t) \urcorner_{B \Rightarrow C}} \\ D & \xrightarrow{\gamma_{\ulcorner \phi'_2(t) \urcorner_{C \Rightarrow D}}} & C \end{array}$$

Notons qu'on a distingué ϕ_1, ϕ_2 de ϕ'_1, ϕ'_2 car ces fonctions dépendent implicitement de l'interaction correspondante. Dans les situations où on pourra alléger un peu ces notations sans provoquer d'ambiguïtés, on n'hésitera bien sûr pas à le faire.

7.2.3 Foncteurs usuels

Aussi bien à titre d'exemple que pour préparer la structure de types forts sur **Par**, prenons le temps de montrer que les foncteurs usuels sont ouverts. Par abus de notation, la notation \square_X utilisée comme arène désignera l'arène réduite à l'unique coup (initial, Opposant) \square_X .

Proposition 7.2.10. *Les foncteurs constants, le foncteur identité, les foncteurs $-_1 \times -_2$, $-_1 + -_2$ et $- \Rightarrow A$ sont ouverts.*

Démonstration. Pour chacun, on va donner l'arène ouverte correspondante. Dans chaque cas, une analyse directe de la forme des stratégies générées par le foncteur montre qu'elles sont bien image du foncteur ouvert correspondant à l'arène ouverte exhibée. Pour les foncteurs constants, il suffit de considérer n'importe quelle arène sans trous comme une arène ouverte. Le foncteur identité est engendré par l'arène ouverte atomique \square_X . Les foncteurs $-_1 \times -_2$ et $-_1 + -_2$ sont engendrés respectivement par l'arène $\square_X \times \square_Y$ et $\square_X + \square_Y$, et $- \Rightarrow A$ est engendré par l'arène $\square_X \Rightarrow A$. \square

Proposition 7.2.11. *La classe des foncteurs ouverts est stable par composition. Plus précisément, si F et G sont deux endofoncteurs ouverts co/contra-variants composables, alors FG est également ouvert.*

Démonstration. Considérons les arènes ouvertes F et G . Construisons alors l'arène $F[G/X]$. On calcule :

$$\begin{aligned} F[G/X](\sigma) &= id_{F[G/X]}[\sigma] \\ &= (id_F[id_G])[\sigma] \\ &= id_F[id_G[\sigma]] \\ &= F(G(\sigma)) \end{aligned}$$

On a utilisé dans l'ordre les Lemmes 7.2.4 et 7.1.19. \square

Proposition 7.2.12. *La classe des foncteurs ouverts est stable par contraction. Plus précisément, si $F(-_1, -_2)$ est un foncteur ouvert, $F(-, -)$ est un foncteur ouvert.*

Démonstration. L'arène ouverte F contient des trous d'étiquettes distinctes X et Y . Il suffit de renommer les Y en X , et le foncteur correspondant sera bien $F(-, -)$. \square

7.3 Théorème de factorisation

Dans la section précédente, on a décrit une classe de foncteurs ouverts et de transformations naturelles entre ces foncteurs. Ici, on va s'intéresser tout particulièrement à ces transformations naturelles ouvertes et montrer à leur sujet un résultat puissant : la classe des transformations naturelles ouvertes est *complète*, c'est-à-dire que toute transformation naturelle entre foncteurs ouverts peut être vue comme une transformation naturelle ouverte. Autrement dit, on va donner pour chaque transformation naturelle $(\eta_A)_A$ une stratégie naturelle η et une factorisation explicite par $\eta_A = \eta[id_A]$. Plus précisément, soient S et T deux foncteurs ouverts tous deux de la même variance fixe, et considérons $\eta : S \xrightarrow{\bullet} T$ une transformation naturelle. La factorisation de η_A est directe : on pose simplement $\eta = \eta_{\square_X}$ où \square_X est vue comme une arène réduite à un point. L'idée est ensuite de montrer que pour toute arène A , on a bien

$\eta_A = \eta[id_A]$. On va montrer ça dans un premier temps pour les forêts finies par induction sur leur structure, pour généraliser aux arènes quelconques par un argument de continuité. Avant de traiter le cas général, commençons par donner la preuve dans le cas où S et T sont tous deux le foncteur identité. Autrement dit, on va montrer que la seule transformation naturelle du foncteur identité dans lui-même est la transformation naturelle identité. Ce cas simple a l'avantage de contenir une grande part du raisonnement, tout en s'affranchissant pour un temps des difficultés techniques liées à la restriction verticale. Ensuite, on fera la preuve dans le cas général.

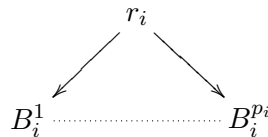
7.3.1 Trivialité pour l'identité

Comme annoncé ci-dessus, nous nous intéressons ici aux transformations naturelles du foncteur identité dans lui-même dans la catégorie des arènes et stratégies innocentes. L'objectif est de démontrer qu'il n'y en a qu'une seule, la transformation identité. Ce n'est pas toujours le cas, voir par exemple les catégories de rubans [JS88]. Dans toute cette sous-section, on fixe une famille $(\iota_A)_A$ naturelle en A , et on va montrer que pour toute arène A , $\iota_A = id_A$.

Lemme 7.3.1 (Factorisation des arènes). *Soit A une arène. Alors A est isomorphe à une arène d'une des formes suivantes :*

- 1 ;
- 0 ;
- $A_1 \times A_2$ avec A_1 et A_2 ayant chacune strictement moins de coups initiaux que A ;
- $\coprod_{i \in I} A_i$;
- $A_1 \Rightarrow A_2$.

Démonstration. Si A est vide, alors $A = 1$. Si A est réduite à un seul coup, alors A est isomorphe à 0. Sinon, on considère le dépliage \tilde{A} de A en forêt (voir Section 2.4.2), avec l'isomorphisme $\gamma_A : A \rightarrow \tilde{A}$ correspondant. Si \tilde{A} a plusieurs coups initiaux, étant donné que c'est une forêt, elle est nécessairement de la forme $A_1 \times A_2$, où les coups initiaux de A_1, A_2 forment une partition arbitraire des coups initiaux de \tilde{A} . Si \tilde{A} n'a qu'un seul coup initial, c'est un arbre. Désignons par $A_1, \dots, A_n, B_1, \dots, B_p$ les sous-arbres de la racine, de sorte que tous les A_i commencent par une question et tous les B_i par une réponse. Si $n > 0$, notons C l'arbre \tilde{A} dans lequel on a enlevé tous les A_i . Alors, \tilde{A} est donné par $\prod_{i \in \{1, \dots, n\}} A_i \Rightarrow C$. Finalement, examinons le cas où $n = 0$, c'est à dire que tous les sous-arbres de la racine de \tilde{A} commencent par une réponse. Notons que par définition des arènes, les réponses ne peuvent justifier que des questions, c'est-à-dire que chaque B_i est de la forme suivante :



où chaque B_i^j est un arbre commençant par une question, *i.e.* une arène bien formée. Alors, c'est que \tilde{A} (donc A) est isomorphe à :

$$\coprod_{i \in \{1, \dots, p\}} \prod_{j \in \{1, \dots, p_i\}} B_i^j$$

□

Lemme 7.3.2 (Commutations aux unités). *On a $\iota_1 = id_1$ et $\iota_0 = id_0$.*

Démonstration. La première égalité est triviale : 1 étant l'arène vide, il n'y a qu'une seule stratégie (laquelle est vide) sur $1 \Rightarrow 1$. Considérons maintenant l'arène \top , constituée d'une question initiale et d'une unique réponse. Considérons également le diagramme de naturalité suivant, qu'on va ensuite instancier pour plusieurs valeurs de σ .

$$\begin{array}{ccc} 0 & \xrightarrow{\iota_0} & 0 \\ \sigma \downarrow & & \downarrow \sigma \\ \top & \xrightarrow{\iota_\top} & \top \end{array}$$

D'abord, remarquons que ι_\top répond nécessairement au premier coup initial. En effet, dans le cas contraire il suffit de prendre σ qui répond immédiatement à droite pour parvenir à une contradiction. Supposons que ι_\top réponde au premier coup initial immédiatement à droite. Alors, on peut trouver une contradiction avec σ qui va immédiatement interroger son argument de gauche. Donc ι_\top va nécessairement interroger son argument. Mais alors c'est que ι_0 va également interroger son argument, sinon on aurait une contradiction avec σ qui joue immédiatement à gauche. Donc $\iota_0 = id_0$. \square

Ce lemme peut être alors étendu au lemme suivant :

Lemme 7.3.3. *Soit A une arène, alors ι_A est stricte, c'est-à-dire qu'elle répond immédiatement à gauche au coup initial d'Opposant.*

Démonstration. Si A est vide c'est évident. Sinon, examinons pour l'unique stratégie $\sigma : 0 \rightarrow A$ stricte, le diagramme de naturalité suivant :

$$\begin{array}{ccc} 0 & \xrightarrow{\iota_0} & 0 \\ \sigma \downarrow & & \downarrow \sigma \\ A & \xrightarrow{\iota_A} & A \end{array}$$

Par le lemme précédent, on sait que le chemin en haut à droite donne l'unique stratégie stricte sur $0 \Rightarrow A$. Par conséquent, ι_A doit être stricte. \square

Lemme 7.3.4 (Commutation au produit). *Soit $A \times B$ une arène, alors*

$$\iota_{A \times B} = \iota_A \times \iota_B$$

Démonstration. Considérons les deux diagrammes de naturalité suivants :

$$\begin{array}{ccc} A \times B & \xrightarrow{\iota_{A \times B}} & A \times B \\ \pi_1 \downarrow & & \downarrow \pi_1 \\ A & \xrightarrow{\iota_A} & A \end{array} \quad \begin{array}{ccc} A \times B & \xrightarrow{\iota_{A \times B}} & A \times B \\ \pi_2 \downarrow & & \downarrow \pi_2 \\ B & \xrightarrow{\iota_B} & B \end{array}$$

La structure cartésienne de **Inn** permet alors de calculer :

$$\begin{aligned} \iota_{A \times B} &= \iota_{A \times B}; \langle \pi_1, \pi_2 \rangle \\ &= \langle \iota_{A \times B}; \pi_1, \iota_{A \times B}; \pi_2 \rangle \\ &= \langle \pi_1; \iota_A, \pi_2; \iota_B \rangle \\ &= \iota_A \times \iota_B \end{aligned}$$

□

Lemme 7.3.5 (Commutation aux sommes). *Soit $\coprod_{i \in I} A_i$ une arène, alors :*

$$\iota_{\coprod_{i \in I} A_i} = \prod_{i \in I} \iota_{A_i}$$

Démonstration. Pour chaque $i \in I$, la naturalité de ι donne le diagramme commutatif suivant :

$$\begin{array}{ccc} A_i & \xrightarrow{\iota_{A_i}} & A_i \\ \text{\scriptsize } in_i \downarrow & & \downarrow \text{\scriptsize } in_i \\ \coprod_{i \in I} A_i & \xrightarrow{\iota_{\coprod_{i \in I} A_i}} & \coprod_{i \in I} A_i \end{array}$$

Ce qui permet le raisonnement équationnel :

$$\begin{aligned} \iota_{\coprod_{i \in I} A_i} &= [in_1, in_2, \dots, in_I]; \iota_{\coprod_{i \in I} A_i} \\ &= [in_1; \iota_{\coprod_{i \in I} A_i}, \dots, in_I; \iota_{\coprod_{i \in I} A_i}] \\ &= [\iota_{A_1}; in_1, \dots, \iota_{A_I}; in_I] \\ &= \prod_{i \in I} \iota_i \end{aligned}$$

Le passage de la première ligne à la seconde utilise le Lemme 2.2.13 (trivialement généralisé aux sommes n -aires) et le fait que $\iota_{\coprod_{i \in I} A_i}$ est stricte par le Lemme 7.3.3. □

Lemme 7.3.6 (Commutation à la flèche). *Soit $A \Rightarrow B$ une arène, et supposons que $\iota_A = id_A$. Alors :*

$$\iota_{A \Rightarrow B} = A \Rightarrow \iota_B$$

Démonstration. Considérons le diagramme de naturalité suivant :

$$\begin{array}{ccc} A \times (A \Rightarrow B) & \xrightarrow{ev} & B \\ \text{\scriptsize } \iota_{A \times (A \Rightarrow B)} \downarrow & & \downarrow \text{\scriptsize } \iota_B \\ A \times (A \Rightarrow B) & \xrightarrow{ev} & B \end{array}$$

Par hypothèse et par les lemmes précédents, on sait que :

$$\begin{aligned} \iota_{A \times A \Rightarrow B} &= \iota_A \times \iota_{A \Rightarrow B} \\ &= A \times \iota_{A \Rightarrow B} \end{aligned}$$

Par conséquent, on a par adjonction la commutation du diagramme suivant :

$$\begin{array}{ccc} A \Rightarrow B & \xrightarrow{id} & A \Rightarrow B \\ \text{\scriptsize } \iota_{A \Rightarrow B} \downarrow & & \downarrow \text{\scriptsize } A \Rightarrow \iota_B \\ A \Rightarrow B & \xrightarrow{id} & A \Rightarrow B \end{array}$$

lequel donne immédiatement l'égalité requise. □

Lemme 7.3.7 (Forêts finies). *Si F est une forêt finie, alors $\iota_F = id_F$.*

Démonstration. Par induction immédiate sur F , et par les lemmes précédents. \square

Ainsi, les règles de commutation nous permettent de montrer le résultat par induction sur une classe d'arène, malheureusement restreinte. On va maintenant voir comment généraliser ça aux forêts infinies, puis finalement aux arènes en général.

Lemme 7.3.8 (Forêts quelconques). *Soit F une forêt (potentiellement infinie), alors $\iota_F = id_F$.*

Démonstration. Soit F' une forêt préfixe de F finie. Comme on l'a vu dans les préliminaires, on dispose de stratégies de projection et d'injection $\mathbf{proj}_{F'} : F \rightarrow F'$ et $\mathbf{inj}_{F'}$. La commutation de ι vis-à-vis de $\mathbf{proj}_{F'}$ donne :

$$\iota_F; \mathbf{proj}_{F'} = \mathbf{proj}_{F'}; \iota_{F'}$$

Or F' est une forêt finie, donc $\iota_{F'} = id_{F'}$. Donc :

$$\iota_F; \mathbf{proj}_{F'} = \mathbf{proj}_{F'}$$

En précomposant cette équation par $\mathbf{inj}_{F'}$, on obtient :

$$\mathbf{inj}_{F'}; \iota_F; \mathbf{proj}_{F'} = id_{F'}$$

Or, la stratégie $\mathbf{inj}_{F'}; \iota_F; \mathbf{proj}_{F'}$ est exactement la restriction de ι_F à la forêt finie F' . Donc pour toute forêt F' finie préfixe de F , on a $\iota_{F'} = id_{F'}$. Par continuité, on a donc $\iota_F = id_F$. \square

Théorème 7.3.9 (Trivialité pour l'identité). *Soit ι une transformation naturelle de l'endofoncteur identité dans lui-même, dans la catégorie des arènes et stratégies innocentes. Alors ι est la transformation naturelle identité.*

Démonstration. Considérons le dépliage de A en forêt éventuellement infinie, noté $\gamma_A : A \rightarrow \tilde{A}$. La naturalité de ι donne le diagramme commutatif suivant :

$$\begin{array}{ccc} A & \xrightarrow{\gamma_A} & \tilde{A} \\ \iota_A \downarrow & & \downarrow \iota_{\tilde{A}} \\ A & \xrightarrow{\gamma_A} & \tilde{A} \end{array}$$

Or \tilde{A} est une forêt, donc $\iota_{\tilde{A}} = id_{\tilde{A}}$. Donc, on a :

$$\iota_A; \gamma_A = \gamma_A$$

Or γ_A est un isomorphisme, d'où on déduit immédiatement le résultat voulu. \square

7.3.2 Factorisation

Ici, on va refaire le travail fait précédemment pour l'identité, mais cette fois dans le cadre plus général où S et T sont distincts du foncteur identité. Pour toute cette sous-section, on fixe deux foncteurs ouverts S et T , ainsi qu'une transformation naturelle $(\eta_A) : S \xrightarrow{\bullet} T$. Notons qu'il est souhaitable que l'argument donné ici s'applique plus tard aux transformations naturelles entre foncteurs ouverts à *gain* dans la catégorie de jeux à gain et stratégies gagnantes qui servira à modéliser μLJ . On ne peut donc s'autoriser que les commutations avec des stratégies qui seront plus tard des stratégies gagnantes (bien sûr, on reviendra sur ce point). Présentons cette preuve dans le cas où S et T sont des foncteurs ouverts covariants, comme d'habitude la preuve dans le cas contravariant est identique.

Lemme 7.3.10. *La stratégie η_{\square_X} est une stratégie naturelle.*

Démonstration. On va commencer par noter quelques propriétés de η_{\top} , pour l'arène \top composée d'une unique question et d'une unique réponse à cette question. Considérons le diagramme commutatif suivant, pour l'unique $\sigma : \top \rightarrow 1$.

$$\begin{array}{ccc} S\top & \xrightarrow{\eta_{\top}} & T\top \\ S\sigma \downarrow & & \downarrow T\sigma \\ S1 & \xrightarrow{\eta_1} & T1 \end{array}$$

L'égalité des deux chemins externes montre que η_{\top} se comporte comme η_1 sur la composante fonctorielle. En particulier, elle ne saute pas de la composante fonctorielle à la composante argument. Remarquons au passage que ce raisonnement est valide pour n'importe quelle arène A . Observons maintenant ce second diagramme :

$$\begin{array}{ccc} S0 & \xrightarrow{\eta_0} & T0 \\ S\sigma \downarrow & & \downarrow T\sigma \\ S\top & \xrightarrow{\eta_{\top}} & T\top \end{array}$$

Si σ répond immédiatement à droite à la question de l'Opposant, ce diagramme montre que η_{\top} doit nécessairement répondre également. Si η_{\top} répond immédiatement à droite, on arrive à une contradiction en prenant σ stricte qui va immédiatement interroger le coup initial sur 0 , ainsi η_{\top} est stricte sur sa composante argument. Examinons maintenant le diagramme suivant, pour une stratégie $\sigma : \square_X \rightarrow \top$ arbitraire.

$$\begin{array}{ccc} S\square_X & \xrightarrow{\eta_{\square_X}} & T\square_X \\ S\sigma \downarrow & & \downarrow T\sigma \\ S\top & \xrightarrow{\eta_{\top}} & T\top \end{array}$$

Ce diagramme montre que η_{\square_X} doit coïncider avec η_{\top} sur la composante foncteur, c'est-à-dire répondre à tout coup de la composante foncteur par un autre coup de la composante foncteur. Mais puisque η_{\top} est «stricte» en \top , on peut fixer σ comme étant la stratégie qui va immédiatement interroger son argument : cela montre que η_{\square_X} doit également être «stricte» en \square_X , et c'est donc bien une stratégie naturelle. \square

On a donc une stratégie naturelle $\eta = \eta_{\square_X}$, qui sera la première composante de la factorisation de $(\eta) : S \xrightarrow{\bullet} T$.

Lemme 7.3.11 (Cas des unités). *On a $\eta_0 = \eta[id_0]$ et $\eta_1 = \eta[id_1]$.*

Démonstration. La première égalité correspond exactement au Lemme 7.3.10 (à renommage de l'unique coup dans 0 près). La seconde égalité s'appuie sur le diagramme de commutation suivant, et l'unique stratégie $\sigma : 0 \rightarrow 1$.

$$\begin{array}{ccc} S0 & \xrightarrow{\eta_0} & T0 \\ S\sigma \downarrow & & \downarrow T\sigma \\ S1 & \xrightarrow{\eta_1} & T1 \end{array}$$

Or, $\eta_0 = \eta[id_0]$ par l'égalité précédente. Notons alors que les parties de η_1 sont celles de $S\sigma$; η_1 , en effet les deux stratégies restent dans la composante fonctorielle, dans laquelle les parties de $S\sigma$ sont celles de l'identité. Mais $S\sigma; \eta_1 = \eta_0$; $T\sigma = \eta[\sigma] = \eta[\epsilon] = \eta[id_1]$, donc on a bien la factorisation espérée. \square

On va maintenant démontrer des lemmes similaires pour tous les constructeurs. Notons qu'on va s'appuyer sur les Lemmes 7.2.7 et 7.2.9, qui permettront de décomposer les diagrammes commutatifs en leur composante fonctorielle et leur composante argument. Pour pouvoir appliquer ces résultats, on a besoin du lemme suivant.

Lemme 7.3.12. *Soit A une arène non vide, alors η_A est stricte sur sa composante argument, c'est-à-dire qu'elle associe un coup initial de A à chaque coup initial de A .*

Démonstration. Il suffit d'examiner le diagramme commutatif suivant, pour $\sigma : A \Rightarrow \top$ qui pose immédiatement un coup initial sur A , puis répond sur \top dès que possible.

$$\begin{array}{ccc} SA & \xrightarrow{\eta_A} & TA \\ S\sigma \downarrow & & \downarrow T\sigma \\ S\top & \xrightarrow{\eta_\top} & T\top \end{array}$$

Comme on sait que η_\top est stricte en sa composante argument, ainsi que $T(\sigma)$ et $S(\sigma)$, ce doit être également le cas de η_A . \square

Lemme 7.3.13 (Cas du produit). *Pour toutes arènes A et B non vides, si $\eta_A = \eta[id_A]$ et $\eta_B = \eta[id_B]$, alors $\eta_{A \times B}$ est une stratégie uniforme, et $\eta_{A \times B} = \eta[id_{A \times B}]$.*

Démonstration. Le raisonnement s'appuie sur la commutation des deux diagrammes suivants :

$$\begin{array}{ccc} S(A \times B) \xrightarrow{\eta_{A \times B}} T(A \times B) & S(A \times B) \xrightarrow{\eta_{A \times B}} T(A \times B) \\ S(\pi_1) \downarrow & \downarrow T(\pi_1) & S(\pi_2) \downarrow & \downarrow T(\pi_2) \\ S(A) \xrightarrow{\eta_A} T(A) & & S(B) \xrightarrow{\eta_B} T(B) \end{array}$$

Ces diagrammes montrent que toute P -vue de $\eta_A = \eta[id_A]$ et toute P -vue de $\eta_B = \eta[id_B]$ sont aussi des P -vues de $\eta_{A \times B}$. L'union disjointe $\eta_A \uplus \eta_B$ est alors toujours une stratégie déterministe, et est totale sur la composante argument, on a alors $\eta_{A \times B} = \eta[id_A] \uplus \eta[id_B]$ puisque $\eta_{A \times B}$ est déterministe. Donc $\eta_{A \times B} = \eta[id_A] \uplus \eta[id_B] = \eta[id_A \times id_B] = \eta[id_{A \times B}]$. \square

Lemme 7.3.14 (Cas de la somme). *Supposons également données des arènes A_1, \dots, A_p telles que pour tout i , $\eta_{A_i} = \eta[id_{A_i}]$. Alors :*

$$\eta_{\coprod_{1 \leq i \leq p} A_i} = \eta[id_{\coprod_{1 \leq i \leq p} A_i}]$$

Démonstration. Preuve similaire à la précédente. On commence par examiner les diagrammes de naturalité vis-à-vis des injections.

$$\begin{array}{ccc} S(\coprod_{1 \leq i \leq p} A_i) & \xrightarrow{\eta_{\coprod_{1 \leq i \leq p} A_i}} & T(\coprod_{1 \leq i \leq p} A_i) \\ S(in_i) \uparrow & & \uparrow T(in_i) \\ S(A_i) & \xrightarrow{\eta_{A_i}} & T(A_i) \end{array}$$

Posons $C = \coprod_{1 \leq i \leq p} A_i$. Justifions que la stratégie η_C est fonctorielle : en effet, chacune de ses parties O -fonctorielles est dans l'une des $S(in_i); \eta_C$, donc elle est fonctorielle si et seulement si toutes ces stratégies sont fonctorielles. Or, le diagramme commutatif ci-dessus permet le calcul suivant :

$$\begin{aligned} S(in_i); \eta_C &= \eta_{A_i}; T(in_i) \\ &= \eta[id_{A_i}]; T(in_i) \\ &= \eta[in_i] \end{aligned}$$

Cette dernière stratégie est obtenue par composition verticale donc est bien fonctorielle. Toutes les stratégies constituant le diagramme commutatif sont donc fonctorielles, ainsi on peut le projeter sur la composante argument, pour chacune de ses tranches. Or les tranches du diagramme sont exactement celles de η_C , puisque $S(in_i)$ se comporte comme l'identité sur la composante fonctorielle. Ainsi, les restrictions du diagramme à la composante argument montrent que sur chaque tranche t , on a :

$$(\eta_C) \upharpoonright_{C \Rightarrow C} = id_{C \Rightarrow C}$$

Ainsi η_C est uniforme, et est bien obtenue par composition verticale. \square

Lemme 7.3.15 (Cas de la flèche). *Soient A et B deux arènes telles que $\eta_A = \eta[id_A]$ et $\eta_B = \eta[id_B]$, alors*

$$\eta_{A \Rightarrow B} = \eta[id_{A \Rightarrow B}]$$

Démonstration. Le raisonnement s'appuie sur le diagramme de naturalité suivant :

$$\begin{array}{ccc} S(A \times (A \Rightarrow B)) & \xrightarrow{\eta_{A \times (A \Rightarrow B)}} & T(A \times (A \Rightarrow B)) \\ S(ev) \downarrow & & \downarrow T(ev) \\ S(B) & \xrightarrow{\eta_B} & T(B) \end{array}$$

Commençons par justifier que $\eta_{A \times (A \Rightarrow B)}$ est fonctorielle. Supposons donc qu'il y ait une partie $s \in \eta_{A \times (A \Rightarrow B)}$ O -fonctorielle mais non bi-fonctorielle. Supposons par l'absurde qu'il

existe un plus petit saut Joueur $s'ab \sqsubseteq^P s$. L'argument repose sur le fait que la P -vue $\lceil s'ab \rceil$ peut être rejouée dans $u \in \eta_{A \times (A \Rightarrow B)} \parallel T(ev)$, créant ainsi un saut dans $\eta_{A \times (A \Rightarrow B)}; T(ev)$ ce qui est impossible par le diagramme ci-dessus. Une telle interaction peut être construite par induction immédiate sur $\lceil s'ab \rceil$. Cela interdit que b soit un saut de type 1, 2 ou 3 : un tel saut serait toujours visible dans $\lceil s'ab \rceil$. Justifions maintenant qu'il n'y a pas de sauts de type 4, supposons par l'absurde qu'il y en ait un. Autrement dit, il existe $s''a''b'' \sqsubseteq^P s'a'b' \sqsubseteq^P s$ tels que b'' et b' soient deux ouvertures d'un coup initial de $A \times (A \Rightarrow B)$ sur $S(A \times (A \Rightarrow B))$, mais avec des pointeurs et adresses différentes. Considérons alors les deux P -vues correspondantes $\lceil s''a''b'' \rceil$ et $\lceil s'a'b' \rceil$. Comme au dessus, ces deux P -vues peuvent être rejouées ensemble dans $u \in \eta_{A \times (A \Rightarrow B)} \parallel T(ev)$, mais on a alors un saut de type 4 dans $\eta_{A \times (A \Rightarrow B)}; T(ev) = S(ev); \eta[id_B]$, ce qui est impossible puisqu'elle est fonctorielle. Ainsi $\eta_{A \times (A \Rightarrow B)}$ est fonctorielle. On peut donc projeter le diagramme de naturalité ci-dessus sur la composante fonctorielle, ce qui montre que $(\eta_{A \times (A \Rightarrow B)})_{\lceil S \Rightarrow T} = \eta$. De même, on peut le projeter sur la composante argument pour chaque tranche de $\eta_{A \times (A \Rightarrow B)}; T(ev)$, qui sont les mêmes que les tranches t de $\eta_{A \times (A \Rightarrow B)}$, pour obtenir pour chaque t le diagramme commutatif suivant :

$$\begin{array}{ccc} A \times (A \Rightarrow B) & \xrightarrow{(\eta_{A \times (A \Rightarrow B)})_{\lceil t A \Rightarrow B}} & A \times (A \Rightarrow B) \\ \text{ev} \downarrow & & \downarrow \text{ev} \\ B & \xrightarrow{id_B} & B \end{array}$$

Par ailleurs, on utilise encore la naturalité de η pour obtenir les deux diagrammes suivants :

$$\begin{array}{ccc} S(A \times (A \Rightarrow B)) \xrightarrow{\eta_{A \times (A \Rightarrow B)}} T(A \times (A \Rightarrow B)) & S(A \times (A \Rightarrow B)) \xrightarrow{\eta_{A \times (A \Rightarrow B)}} T(A \times (A \Rightarrow B)) \\ \downarrow S(\pi_1) & \downarrow T(\pi_1) & \downarrow S(\pi_2) & \downarrow T(\pi_2) \\ S(A) \xrightarrow{\eta_A = \eta[id_A]} T(A) & & S(A \Rightarrow B) \xrightarrow{\eta_{A \Rightarrow B}} T(A \Rightarrow B) \end{array}$$

Comme on sait déjà que $\eta_{A \times (A \Rightarrow B)}$ est fonctorielle, on peut projeter ces diagrammes sur la composante argument pour chaque tranche t de $\eta_{A \times (A \Rightarrow B)}$. On obtient ainsi la décomposition :

$$\begin{aligned} (\eta_{A \times (A \times B)})_{\lceil t A \Rightarrow B} &= (\eta_{A \times (A \times B)})_{\lceil t A \Rightarrow B}; \langle \pi_1, \pi_2 \rangle \\ &= \langle (\eta_{A \times (A \times B)})_{\lceil t A \Rightarrow B}; \pi_1, (\eta_{A \times (A \times B)})_{\lceil t A \Rightarrow B}; \pi_2 \rangle \\ &= \langle \pi_1, \pi_2; (\eta_{A \Rightarrow B})_{\lceil t A \Rightarrow B} \rangle \end{aligned}$$

Il n'y a plus qu'à effectuer le calcul suivant, avec l'égalité ci-dessus et le diagramme d'apara-
vant :

$$\begin{aligned} (\eta_{A \Rightarrow B})_{\lceil t A \Rightarrow B} &= \Lambda(\langle \pi_1, \pi_2; (\eta_{A \Rightarrow B})_{\lceil t A \Rightarrow B} \rangle; ev) \\ &= \Lambda((\eta_{A \times (A \times B)})_{\lceil t A \Rightarrow B}; ev) \\ &= \Lambda(ev) \\ &= id_{A \Rightarrow B} \quad \square \end{aligned}$$

Lemme 7.3.16 (Forêts finies). *Soit F une forêt finie, alors $\eta_F = \eta[id_F]$.*

Démonstration. Induction immédiate sur la structure des arènes. □

Lemme 7.3.17 (Forêts quelconques). *Si F est une forêt éventuellement infinie, alors $\eta_F = \eta[id_F]$.*

Démonstration. On considère une chaîne croissante de forêts finies préfixes de F , par exemple la chaîne $F_1 \trianglelefteq F_2 \trianglelefteq F_3 \trianglelefteq \dots$ des préfixes de profondeur n . On a alors :

$$\bigsqcup_{i=1}^{+\infty} F_i = F$$

Pour chaque i , le diagramme suivant commute par naturalité de η .

$$\begin{array}{ccc} S(F) & \xrightarrow{\eta_F} & T(F) \\ S(\mathbf{proj}_i) \downarrow & & \downarrow T(\mathbf{proj}_i) \\ S(F_i) & \xrightarrow{\eta_{F_i}} & T(F_i) \end{array}$$

En utilisant le Lemme 7.3.16, ce diagramme se réécrit comme suit :

$$\begin{array}{ccc} S(F) & \xrightarrow{\eta_F} & T(F) \\ S(\mathbf{proj}_i) \downarrow & & \downarrow T(\mathbf{proj}_i) \\ S(F_i) & \xrightarrow{\eta[id_{F_i}]} & T(F_i) \end{array}$$

Maintenant remarquons que les familles $S(\mathbf{proj}_i)$, $T(\mathbf{proj}_i)$ et $\eta[id_{F_i}]$ sont des chaînes strictement croissantes, or on sait que **Inn** est une catégorie enrichie d'ordres complets, donc la composition commute à la limite. On a donc la commutativité du diagramme limite de ceux ci-dessus, qui est :

$$\begin{array}{ccc} S(F) & \xrightarrow{\eta_F} & T(F) \\ id_{S(F)} \downarrow & & \downarrow id_{T(F)} \\ S(F) & \xrightarrow{\eta[id_F]} & T(F) \end{array}$$

Ce qui conclut l'argument. □

Théorème 7.3.18 (Factorisation). *Soit $\eta : S \xrightarrow{\bullet} T$ une transformation naturelle. Alors pour toute arène A , on a $\eta_A = \eta[id_A]$.*

Démonstration. Comme pour le cas de l'identité. On prend le dépliage \tilde{A} de A et l'isomorphisme $\gamma_A : A \rightarrow \tilde{A}$ correspondant. Par hypothèse sur η , on a :

$$\begin{aligned} \eta_A &= \eta_A; T(\gamma_A); T(\gamma_A^{-1}) \\ &= S(\gamma_A); \eta_{\tilde{A}}; T(\gamma_A^{-1}) \\ &= S(\gamma_A); \eta[id_{\tilde{A}}]; T(\gamma_A^{-1}) \\ &= \eta[\gamma_A; \gamma_A^{-1}] \\ &= \eta[id_A] \quad \square \end{aligned}$$

Donc, toute transformation naturelle entre foncteurs ouverts est bien une transformation naturelle ouverte.

7.4 Transformations dinaturelles ouvertes

7.4.1 Échange dinaturel

À la Section 7.2.1, on a vu que les stratégies naturelles sur les arènes de la forme $S \Rightarrow T$, où S et T sont deux arènes ouvertes de même variance fixe, donnent lieu à des transformations naturelles. Mais le lecteur aura noté que la notion de composition verticale a été définie dans le cadre plus général d'une arène de la forme $S[B, A] \Rightarrow T[A, B]$ où on n'a pas cette hypothèse sur la variance de S et T . Dès lors, il devient naturel de se demander si une stratégie naturelle sur une telle arène donne lieu à une transformation dinaturelle entre les foncteurs ouverts à variance mixte correspondants. Comme on va le voir dans cette section, c'est bien le cas.

Lemme 7.4.1 (Échange dinaturel). *Soit $\eta : S \Rightarrow T$ une stratégie naturelle, et soient $\sigma : A \Rightarrow B$, $\delta : B \Rightarrow C$ et $\tau : C \Rightarrow D$ des stratégies innocentes. Alors :*

$$\eta[\sigma; \delta; \tau] = S(\tau, \sigma); \eta[\delta]; T(\sigma, \tau)$$

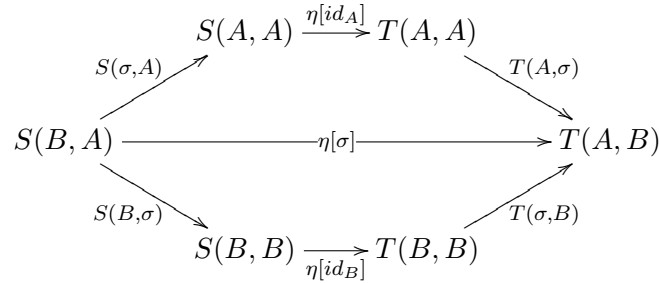
Démonstration. Montrons les deux inclusions sur les parties bi-fonctorielles. Soit $s \in \eta[\sigma; \delta; \tau]$, on va construire un témoin pour s dans l'interaction $I(S(D, A), S(C, B), T(B, C), T(A, D))$. On commence par décomposer s en $s_1 = s_{\uparrow S \Rightarrow T} \in \eta$ et $s_2 = s_{\uparrow A \Rightarrow D} \in \sigma; \delta; \tau$. Mais on a également $s \in id_S; \eta; id_T$, donc il existe deux interactions quaternaires $u_1 \in id_S \parallel \eta \parallel id_T$ et $u_2 \in \sigma \parallel \delta \parallel \tau$, respectivement pour s_1 et s_2 . Si s reste dans la composante argument, u_1 convient. Sinon, quelles que soient les composantes visitées de A, B, C, D dans u_2 , comme η, id_S, id_T est naturelle, il existe une unique façon d'arranger u_1 en u'_1 pour qu'elle comporte exactement les occurrences de \square_X correspondant aux occurrences de coups initiaux de la composante argument dans u_2 . Dans ce cas, u'_1 finit par une séquence d'occurrences de \square_X , chacune correspondant à une des composantes parmi A, B, C, D . On définit alors u en concaténant u'_1 et u'_2 après avoir enlevé cette séquence, et en faisant pointer les coups initiaux de u_2 de la même façon que les occurrences de \square_X correspondant. Une vérification directe montre qu'on a bien $u \in S(\tau; \sigma) \parallel \eta[\delta] \parallel T(\sigma, \tau)$ et que c'est un témoin pour s .

Dans l'autre sens, soit $s \in S(\tau; \sigma); \eta[\delta]; T(\sigma, \tau)$. Soit $u \in S(\tau; \sigma) \parallel \eta[\delta] \parallel T(\sigma, \tau)$ son témoin. Comme s était bi-fonctorielle, u est nécessairement de la forme $u_1 u_2$ avec u_1 dans la composante fonctorielle et u_2 dans la composante argument. Par le Lemme 7.1.4, tous les coups initiaux de la composante $X \in \{A, B, C, D\}$ pointent vers le même coup de la composante fonctorielle et partagent le même préfixe d'adresse, on définit alors u'_1 comme u_1 suivi d'occurrences de \square_X correspondant aux coups initiaux rencontrés de D, C, B, A (dans cet ordre), avec les pointeurs et adresses correspondantes. On a alors par construction $u'_1 \in id_S \parallel \eta \parallel id_T$ et u'_1 témoin pour $s_{\uparrow S \Rightarrow T}$, donc $s_{\uparrow S \Rightarrow T} \in \eta$. De même, par le Lemme 7.1.9, il existe une unique façon de réarranger les pointeurs de u_2 en u'_2 de sorte que ce soit une interaction correcte sur $I(A, B, C, D)$, c'est-à-dire que chaque projection forme une partie légale. On a alors par construction $u'_2 \in \sigma \parallel \delta \parallel \tau$, et u'_2 est bien un témoin pour $s_{\uparrow A \Rightarrow D}$, donc $s_{\uparrow A \Rightarrow D} \in \sigma; \delta; \tau$, ce qui conclut la preuve. \square

De cet échange dinaturel, on déduit immédiatement la propriété requise :

Proposition 7.4.2. *Soit η une stratégie naturelle sur $S \Rightarrow T$. Alors, la famille $\eta_A = \eta[id_A]$ est dinaturelle en A .*

Démonstration. Deux applications de l'échange dinaturel, pour mettre en relation chaque chemin du diagramme de dinaturalité au morphisme central.



□

7.4.2 Stabilité par composition

Les stratégies naturelles donnant lieu à des transformations naturelles, c'est-à-dire celles sur une arène de la forme $S \Rightarrow T$, où S et T sont à variance fixe identique, sont stables par composition. C'est évident, car si $\eta : S \Rightarrow T$ et $\epsilon : T \Rightarrow U$ sont deux stratégie naturelle, alors chacune «change de côté» à chaque apparition d'une occurrence de \square_X : ce sera donc aussi le cas de leur composée $\eta; \epsilon$ qui sera également naturelle. La situation est plus complexe si S et T n'ont pas cette forme restreinte. En effet, on peut observer des effets similaires à celui causant les ruptures de totalité dans la Partie I : η et ϵ peuvent s'échanger des occurrences de \square_X sur l'arène centrale à l'infini, sans qu'une telle occurrence n'arrive un jour sur les arènes «externes» S ou U , brisant ainsi la condition de naturalité pour $\eta; \epsilon$.

Notons que cette possibilité n'est que théorique. De par la forme restreinte des arènes ouvertes considérées (le fait que les trous ne justifient personne), η et ϵ ne peuvent en fait s'échanger les occurrences du trou qu'un nombre fini de fois, et leur composition sera bien naturelle.

Lemme 7.4.3. *Soient T une arène ouverte, et soit $s \in \mathcal{L}_T$ ultimement visible ($j < i$ est son indice de visibilité) éventuellement infinie, telle que si $s_i = \square_X$, alors tous les coups suivants sont également des trous. Alors, si s contient une occurrence de \square_X , s est finie. En particulier, s ne peut pas «finir» par une infinité d'occurrences de \square_X .*

Démonstration. Supposons qu'il existe $s_i = \square_X$ (on notera ici $s_i = \square_X^i$ pour le distinguer d'autres occurrences de \square_X) dans s . C'est donc qu'on peut voir s sous la forme :

$$\dots\dots s_j \overset{\frown}{\dots\dots} \square_X^i$$

Mais plus personne ne pointera jamais plus sur \square_X par contrainte sur les arènes ouvertes et s est visible, donc on peut appliquer le Lemme 4.2.15 à la structure de pointeurs sous-jacente, ce qui nous informe en particulier que personne ne pointera plus jamais dans l'intervalle $[s_j, \dots, \square_X]$. Si s_i est le dernier coup de s , le résultat est montré. Sinon, par hypothèse, c'est forcément encore une occurrence de \square_X . Mais dans ce cas, on vient de montrer que la partie sera de la forme :

$$\dots\dots s_k \overset{\frown}{\dots\dots} s_j \overset{\frown}{\dots\dots} \square_X^i \quad \square_X^{i+1}$$

On peut alors itérer le raisonnement, mais on a nécessairement $k < j$, donc s se termine nécessairement en moins de k étapes. □

Remarque. Si on avait une classe de foncteurs ouverts plus complète, avec des trous justifiant d'autres coups, ce lemme perdrait nécessairement en généralité. On pourrait alors fort bien imaginer deux stratégies naturelles s'échangeant des trous jusqu'à l'infini : ce genre de divergence est à mettre en relation avec le fait qu'en règle générale, les transformations dinaturelles ne sont pas stables par composition. Il serait alors tout à fait possible de se restreindre à des stratégies $\eta : S \Rightarrow T$ naturelles qui soient en plus finies, bornées ou noëthériennes : les théorèmes de finitude de la Partie I interdiraient alors cette divergence, et la classe des transformations dinaturelles ouvertes correspondante serait stable par composition et parfaitement légitime pour construire des modèles du polymorphisme, par exemple du système F.

Lemme 7.4.4. *Les stratégies naturelles sont stables par composition.*

Démonstration. Soient $\eta : S \Rightarrow T$ et $\epsilon : T \Rightarrow U$ deux stratégies naturelles, et soit $sab \in \eta; \sigma$. Considérons un témoin $uau'b \in \eta || \epsilon$ pour sab . Si $a = \square_X$ ou $b = \square_X$, on déduit par induction immédiate sur u' que tous les coups de u' sont des occurrences de \square_X et que a et b en sont tous deux. De plus, si $s \in \eta; \epsilon$ et si a est une occurrence de \square_X telle que sa est une partie légale, alors s'il n'y a pas b telle que $sab \in \eta; \epsilon$, on peut générer par naturalité une interaction u infinie ultimement visible témoin pour sa dans $\eta || \epsilon$, qui devra nécessairement se terminer par une queue infinie d'occurrences de \square_X par naturalité de η et ϵ . Alors, sa restriction à T violerait immédiatement le Lemme 7.4.3. \square

Lemme 7.4.5 (Compatibilité avec la composition). *Soient $\eta : S \Rightarrow T$ et $\epsilon : T \Rightarrow U$ deux stratégies naturelles, alors pour toute arène A ,*

$$(\eta; \epsilon)[id_A] = \eta[id_A]; \epsilon[id_A]$$

Démonstration. Même méthode de preuve que pour les propositions similaires précédentes, sauf que cette fois on ne sait pas à l'avance combien de fois η et ϵ vont se «renvoyer la balle» sur A . Pour gérer cette subtilité, on va utiliser la notion d'interaction n -aire. Si $n > 2$, on note $I_n(A)$ les parties u sur n copies de A notées A_1, \dots, A_n telles que :

- $\forall i \in \{1, \dots, n-1\}, su_{\uparrow A_i, A_{i+1}} \in \mathcal{L}_{A \Rightarrow A}$
- $s_{\uparrow A_1 \Rightarrow A_n} \in \mathcal{L}_{A \Rightarrow A}$

Passons maintenant à la démonstration proprement dite. Comme d'habitude, montrons la double inclusion sur les parties bi-fonctorielles. Soit donc $s \in (\eta; \epsilon)[id_A]$ bi-fonctorielle. Soit s n'atteint pas A , auquel cas il n'y a rien à montrer, soit s atteint A , et atteint dans ce cas nécessairement les deux composantes. Dans ce cas, on pose $s_1 = s_{\uparrow S \Rightarrow U} \in \eta; \epsilon$ qui finit nécessairement par deux occurrences de \square_X , et $s_2 = s_{\uparrow A \Rightarrow A} \in id_A$. Comme $s_1 \in \eta; \epsilon$, s_1 admet un témoin $u_1 \in \eta || \epsilon$ bi-fonctoriel sur toutes ses composantes. Comme s_1 finit par deux occurrences de \square_X , u_1 finit lui par $n > 2$ occurrences de \square_X . Mais alors, $s_2 \in id_A$ génère de façon unique une interaction $u_2 \in I_n(A)$. On construit alors comme d'habitude $u = u'_1 u'_2$, où u'_1 est obtenu en supprimant les occurrences de \square_X de u_1 et u'_2 est obtenu en faisant pointer les coups initiaux de A dans u_2 là où pointaient les occurrences de \square_X , en identifiant les coups initiaux aux occurrences de \square_X dans l'ordre chronologique. Ainsi, on obtient bien une interaction $u \in I(S[A, A], T[A, A], U[A, A])$. On a également par construction que $u \in \eta[id_A] || \epsilon[id_A]$, et que u est un témoin pour s .

Soit maintenant $s \in \eta[id_A]; \epsilon[id_A]$ bi-fonctorielle. Soit $u \in \eta[id_A] || \epsilon[id_A]$ un témoin pour s . Par bi-fonctorialité, u est de la forme $u_1 u_2$, avec u_1 dans la composante fonctorielle et u_2

dans la composante argument. On obtient alors u'_1 en concaténant à u_1 des copies de \square_X , une pour chaque copie de A traversée, avec le pointeur commun des coups initiaux sur cette copie (le fait que ces coups initiaux aient un pointeur commun est assuré par le Lemme 7.1.4). On a alors $u'_1 \in \eta \parallel \epsilon$, et par construction $u'_{1S[A,A],U[A,A]} = s_{\uparrow S \Rightarrow U}$. De même, une analyse directe de u_2 montre que $s_{\uparrow A \Rightarrow A} = id_A$, comme projection externe d'une interaction n -aire entre copies de l'identité sur A . \square

Notons qu'on peut associer ce lemme avec la loi d'échange dinaturelle, les deux permettant en effet de faire des manipulations algébriques sur l'application des transformations dinaturelles ouvertes. Dans le cas naturel, la loi d'échange couvre en un seul coup ces deux types de manipulation. Cependant, le cas dinaturel rend délicate la formulation d'un lemme permettant directement ces deux manipulations, entre autres parce qu'une stratégie σ appliquée dans la composante argument peut être «renvoyée» un nombre arbitraire (mais fini) de fois entre η et ϵ , avant de ressortir à l'extérieur de l'interaction.

7.4.3 Sur la factorisation.

Mentionnons brièvement un corollaire immédiat de la sous-section précédente :

Corollaire 7.4.6. *Les transformations dinaturelles ouvertes sont stables par composition.*

Démonstration. Conséquence directe des Lemmes 7.4.4 et 7.4.5. \square

On a donc ici une classe de transformations dinaturelles entre foncteurs ouverts qui est stable par composition. Par ailleurs, on a montré dans la section précédente que toutes les transformations naturelles entre foncteurs ouverts étaient en fait obtenues de cette façon, il est donc légitime de se demander si la situation est identique dans le cas des transformations dinaturelles. Ce résultat pourrait être intéressant, par exemple pour modéliser le polymorphisme. Malheureusement cette question ne sera pas traitée ici. En effet, ce résultat de factorisation n'est pas nécessaire pour le projet qui nous occupe, et semble s'avérer techniquement plus difficile que dans le cas des transformations naturelles.

7.5 Types Forts

Les outils introduits dans les sections précédentes vont maintenant être appliqués pour équiper la catégorie **Inn** d'une structure de *types forts*. Comme requis, on a déjà une classe de foncteurs contenant les constructeurs usuels, et stable par composition et par contraction : les foncteurs ouverts. Il faut maintenant donner une *force* à chacun de ces foncteurs ; on les construira comme transformations (di)naturelles ouvertes. Dans la première sous-section, on donnera la définition des forces et contraforces. Elles seront évidemment (di)naturelles par construction, mais il faudra montrer qu'elles vérifient bien les diagrammes de cohérence nécessaires. Dans la seconde sous-section, on vérifiera qu'elles donnent une structure de types forts, c'est-à-dire qu'elles sont compatibles entre elles.

7.5.1 Foncteurs ouverts et forces

Foncteurs covariants. Les foncteurs ouverts covariants sont forts.

Définition 7.5.1. Soit $T : \mathbf{Inn} \rightarrow \mathbf{Inn}$ un foncteur ouvert covariant. On définit une stratégie $\theta^T : \square_X \times T(\square_Y) \rightarrow T(\square_X \times \square_Y)$ naturelle en X et Y par :

$$\theta^T = \{s \in \mathcal{L}_{\square_X \times T(\square_Y) \Rightarrow T(\square_X \times \square_Y)} \mid \forall s' \sqsubseteq^P s, s'_{\{L\}} \simeq s'_{\{R\}}\}$$

Par $s \simeq s'$, on entend ici que les deux parties coïncident, au pointeur d'éventuelles occurrences de \square_X près. C'est une stratégie sans mémoire, donc innocente.

Définition 7.5.2. On définit la force $\theta_{\Gamma, A}^T : \Gamma \times T(A) \rightarrow T(\Gamma \times A)$ de T par composition verticale :

$$\theta_{\Gamma, A}^T = \theta^T[id_{\Gamma}/X, id_A/Y]$$

Par la loi d'échange, c'est une transformation naturelle en Γ et en A .

Proposition 7.5.1. Si $T : \mathbf{Inn} \rightarrow \mathbf{Inn}$ est un foncteur ouvert alors $\theta_{\Gamma, A}^T$ est une force.

Démonstration. La cohérence vis-à-vis des unités est triviale, montrons la cohérence vis-à-vis de l'associativité. Appliquons ce qu'on sait sur les transformations naturelles ouvertes. On décompose, en se servant des différentes manipulations algébriques connues sur les transformations naturelles ouvertes et du théorème de factorisation des transformations naturelles.

$$\begin{aligned} \theta_{A \times B, C}^T &= \theta^T[id_{A \times B}/X, id_C/Z] \\ &= \theta^T[id_{\square_X \times \square_Y}[id_A/X, id_B/Y]/X, id_X/Z] \\ &= (\theta^T[\square_X \times \square_Y/X])[id_A/X, id_B/Y, id_C/Z] \end{aligned}$$

$$\alpha_{A, B, C} = \alpha[id_A/X, id_B/Y, id_C/Z]$$

$$\begin{aligned} \alpha_{A, B, T(C)} &= \alpha[id_A/X, id_B/Y, id_{T(C)}/Z] \\ &= \alpha[T(\square_Z)/Z][id_A/X, id_B/Y, id_C/Z] \end{aligned}$$

Et suivant le même processus, avec un peu moins de détails :

$$id_A \times \theta_{B, C}^T = (id_{\square_X} \times \theta^T[\square_Y/X, \square_Z/Y])[id_A/X, id_B/Y, id_C/Z]$$

$$\theta_{A, B \times C}^T = (\theta[\square_Y \times \square_Z/Y])[id_A/X, id_B/Y, id_C/Z]$$

Ainsi, on décompose chaque chemin du diagramme en la composition verticale d'une stratégie naturelle et des identités sur A, B et C . Par la loi d'échange, la commutativité du diagramme pour tous A, B et C est alors conséquence de celle pour $A = \square_X, B = \square_Y$ et $C = \square_Z$. Or, ces deux chemins donnent chacun une stratégie sur

$$(\square_X \times \square_Y) \times T(\square_Z) \Rightarrow T(\square_X \times (\square_Y \times \square_Z))$$

et une application directe des définitions de θ et α montrent que ces deux chemins sont l'identité sur T . Comme ce sont des stratégies naturelles, ils doivent nécessairement associer l'occurrence de droite d'un trou à celle de gauche correspondante, d'où leur égalité. \square

Foncteurs contravariants. Les foncteurs ouverts contravariants sont également forts.

Définition 7.5.3. Soit $N : \mathbf{Inn}^{op} \rightarrow \mathbf{Inn}$ un foncteur ouvert contravariant. On définit une stratégie $\rho^N : \square_X \times N(\square_X \times \square_Y) \rightarrow N(\square_Y)$ par :

$$\rho^N = id_{N(\square_Y)} \cup \{s \square_X \square_X \in \mathcal{L}_{\square_X \times N(\square_X \times \square_Y) \rightarrow N(\square_Y)} \mid s \in id_{N(\square_Y)}\}$$

Autrement dit, ρ^N se comporte comme l'identité sur N , puis associe l'occurrence Opposant de chaque trou à son occurrence Joueur. C'est une stratégie sans mémoire, donc innocente.

Définition 7.5.4. On définit la contraforce $\rho_{\Gamma,A}^N : \Gamma \times N(\Gamma \times A) \rightarrow N(A)$ de N par la composition verticale suivante :

$$\rho_{\Gamma,A}^N = \rho^N[id_{\Gamma}/X, id_A/Y]$$

Par les lois d'échange naturel et dinaturel, c'est une transformation naturelle en A et dinaturelle en Γ .

Proposition 7.5.2. Si $N : \mathbf{Inn}^{op} \rightarrow \mathbf{Inn}$ est un foncteur ouvert, alors $\rho_{\Gamma,A}^N$ est une contraforce.

Démonstration. Encore une fois, le premier diagramme de cohérence (unités) est trivial. Pour le second diagramme, on vérifie que celui-ci commute pour $A = \square_X$, $B = \square_Y$, $C = \square_Z$: c'est simple par la définition des différentes stratégies composant le diagramme, qui se comportent toutes comme l'identité sur N et associent chaque trou Opposant au trou Joueur correspondant. Puis, on conclut par les lois d'échange et les diverses manipulations algébriques disponibles pour la composition verticale. \square

7.5.2 Structure de types forts

Ainsi tous les foncteurs ouverts, qu'ils soient covariants ou contravariants, sont équipés d'une force. Il reste à montrer que ces forces sont uniformes, c'est-à-dire qu'elles définissent une structure de types forts. En particulier, il nous faut vérifier que les forces vérifient les conditions de compatibilité de la Définition 6.2.5. Vérifions-les une à une.

Lemme 7.5.3. Les familles $\theta_{\Gamma,A}^T$ et $\rho_{\Gamma,A}^N$ sont naturelles respectivement en T et N .

Démonstration. Pour commencer, montrons-le pour θ . Soit $\eta_A : T_1 A \rightarrow T_2 A$ une transformation naturelle de T_1 vers T_2 . Il faut montrer que pour tous Γ, A , le diagramme suivant commute :

$$\begin{array}{ccc} \Gamma \times T_1(A) & \xrightarrow{\theta_{\Gamma,A}^{T_1}} & T_1(\Gamma \times A) \\ \Gamma \times \eta_A \downarrow & & \downarrow \eta_{\Gamma \times A} \\ \Gamma \times T_2(A) & \xrightarrow{\theta_{\Gamma,A}^{T_2}} & T_2(\Gamma \times A) \end{array}$$

Par le théorème de factorisation, on a $\eta_A = \eta[id_A]$. Comme précédemment, on s'intéresse au cas où $\Gamma = \square_X$ et $A = \square_Y$, qui permet de démontrer le cas général par les manipulations algébriques sur la composition verticale. Or l'égalité est directe pour ce cas particulier, les deux chemins donnant la stratégie qui se comporte comme η sur la composante fonctorielle et associe chaque occurrence Opposant d'un trou à l'occurrence Joueur correspondante. La preuve est similaire pour le cas contravariant. \square

Lemme 7.5.4. *La famille θ est compatible avec l'identité et les foncteurs constants :*

$$\begin{aligned}\theta_{\Gamma,A}^- &= id_{\Gamma \times A} \\ \theta_{\Gamma,A}^B &= \pi_2 \\ \rho_{\Gamma,A}^B &= \pi_2\end{aligned}$$

Démonstration. Trivial par définition de θ . □

Lemme 7.5.5. *Les (contra)forces θ et ρ sont compatibles par composition.*

- Si F et G sont covariants :

$$\theta_{\Gamma,A}^{FG} = \theta_{\Gamma,G(A)}^F; F(\theta_{\Gamma,A}^G)$$

- Si F est covariant et G est contravariant :

$$\rho_{\Gamma,A}^{FG} = \theta_{\Gamma,G(\Gamma \times A)}^F; F(\rho_{\Gamma,A}^G)$$

- Si F est contravariant et G est covariant :

$$\rho_{\Gamma,A}^{FG} = \Gamma \times F(\theta_{\Gamma,A}^G); \rho_{\Gamma,G(A)}^F$$

- Si F et G sont contravariants :

$$\theta_{\Gamma,A}^{FG} = \Gamma \times F(\rho_{\Gamma,A}^G); \rho_{\Gamma,G(\Gamma \times A)}^F$$

Démonstration. Pour chacun des quatre cas, on applique la même méthode que précédemment. On montre les équations pour $\Gamma = \square_X$ et $A = \square_Y$, qui sont simples car les forces et contraforces se comportent comme l'identité sur F et G et associent à chaque trou l'unique autre trou de même nom possible. Le résultat s'en déduit par manipulation algébrique sur les foncteurs ouverts. Notons que dans l'élaboration de cette structure de types forts, on utilise pour la première fois le Lemme 7.4.5, qui est indispensable puisqu'on raisonne sur des compositions de transformations dinaturelles ouvertes. □

Lemme 7.5.6. *θ et ρ sont compatibles par contraction, c'est-à-dire, pour P covariant et N contravariant :*

$$\begin{aligned}\theta_{\Gamma,A}^{P(-,-)} &= \langle \pi_1, \theta_{\Gamma,A}^{P(-,A)} \rangle; \theta_{\Gamma,A}^{P(\Gamma \times A,-)} \\ \rho_{\Gamma,A}^{N(-,-)} &= \langle \pi_1, \rho_{\Gamma,A}^{N(-,\Gamma \times A)} \rangle; \rho_{\Gamma,A}^{N(A,-)}\end{aligned}$$

Démonstration. Même idée de preuve. Pour ces deux équations, les deux côtés de l'équation coïncident pour le cas $\Gamma = \square_X$ et $A = \square_Y$, ce qui implique le cas général. □

Lemme 7.5.7. *θ est compatible avec les sommes, c'est-à-dire :*

$$\theta_{\Gamma,A}^{-+-} = \gamma_{\Gamma,A,A}$$

Démonstration. Par le même raisonnement, pour prouver l'égalité de ces deux transformations naturelles il suffit de la montrer avec $\Gamma = \square_X$ et $A = \square_Y$, mais c'est direct. □

Lemme 7.5.8. θ et ρ sont compatibles avec la structure cartésienne fermée, c'est-à-dire :

$$\rho_{\Gamma, A}^{-\Rightarrow C} = \Lambda(\langle \langle \pi_2; \pi_1, \pi_1 \rangle, \pi_2; \pi_2 \rangle; ev)$$

Démonstration. Comparons plutôt leurs formes curryfiées $\Lambda(\rho_{\Gamma, A}^{-\Rightarrow C}) : (\Gamma \times A) \Rightarrow C \rightarrow \Gamma \Rightarrow (A \Rightarrow C)$ et $\Lambda(\Lambda(\langle \langle \pi_2; \pi_1, \pi_1 \rangle, \pi_2; \pi_2 \rangle; ev)) : (\Gamma \times A) \Rightarrow C \rightarrow \Gamma \Rightarrow (A \Rightarrow C)$. Un calcul direct sur la structure cartésienne fermée de **Inn** et la naturalité de ρ en Γ et A montre que ces deux termes sont en fait naturels en Γ et A , donc pour prouver l'égalité, il suffit de la montrer pour $\Gamma = \square_X$ et $A = \square_Y$. Par définition de la curryfication, il suffit donc de montrer l'égalité de l'énoncé, sans la curryfication supplémentaire, pour $\Gamma = \square_X$ et $A = \square_Y$. Or une analyse directe montre que ces stratégies comportent les mêmes parties, qui simulent l'identité sur C et qui, pour chaque trou, associent immédiatement l'occurrence Joueur à l'occurrence Opposant. \square

Théorème 7.5.9. La catégorie **Inn** a les types pseudo-polynomiaux forts.

Démonstration. La famille de foncteurs nécessaire est formée par les foncteurs ouverts. On a montré ci-dessus qu'ils étaient forts, et que leurs forces vérifiaient les équations nécessaires. \square

7.5.3 Foncteur ouverts et types récurifs

Dans cette brève sous-section, on va montrer quelques propriétés de l'action des foncteurs ouverts vis-à-vis de l'ordre \trianglelefteq : en particulier qu'ils sont croissants et continus, et qu'ils préservent les inclusions et projections correspondantes. Étant donné que ces foncteurs sont forts (ce qui correspond à la notion de McCusker de foncteurs *fermés*), les foncteurs ouverts vérifient bien les conditions données dans les préliminaires, assurant que leur itération infinie est bien définie et qu'elle donne des invariants minimaux. Cette sous-section n'aura que peu d'importance technique dans les développements suivants, aussi on se restreindra pour simplifier au cas de foncteurs ouverts $T : \mathbf{Inn} \rightarrow \mathbf{Inn}$.

Lemme 7.5.10. Soit $T : \mathbf{Inn} \rightarrow \mathbf{Inn}$ un foncteur ouvert, alors T est croissante et continue vis-à-vis de \trianglelefteq .

Démonstration. Immédiat par définition de la substitution d'arènes. \square

Lemme 7.5.11. Soit $T : \mathbf{Inn} \rightarrow \mathbf{Inn}$ un foncteur ouvert, alors T préserve les inclusions et projections. Plus formellement si $A \trianglelefteq B$, considérons les morphismes d'inclusion et projection $\mathbf{inj}_{A \trianglelefteq B} : A \rightarrow B$ et $\mathbf{proj}_{A \trianglelefteq B} : B \rightarrow A$, définis comme l'identité restreinte à ces domaines et codomaines respectifs. On a :

$$\begin{aligned} T(\mathbf{inj}_{A \trianglelefteq B}) &= \mathbf{inj}_{T(A) \trianglelefteq T(B)} \\ T(\mathbf{proj}_{A \trianglelefteq B}) &= \mathbf{proj}_{T(A) \trianglelefteq T(B)} \end{aligned}$$

Démonstration. Montrons la première égalité, l'autre est similaire. Soit $s \in T(\mathbf{inj}_{A \trianglelefteq B})$. En particulier, $s \in T(id_A) = id_{T(A)}$, donc $s \in \mathbf{inj}_{T(A) \trianglelefteq T(B)}$. De même, si $s \in \mathbf{inj}_{T(A) \trianglelefteq T(B)}$, on a $s \in id_{T(A)} = T(id_A)$ donc $s \in T(\mathbf{inj}_{A \trianglelefteq B})$. \square

Lemme 7.5.12. Soit $T : \mathbf{Inn} \rightarrow \mathbf{Inn}$, alors T est fermé au sens de McCusker (voir la Section 2.4.1).

Démonstration. Pour toutes arènes A, B on définit :

$$f_{A,B}^T = \Lambda(\langle \pi_2, \pi_1 \rangle; \theta_{A \Rightarrow B, A}^T; T(\langle \pi_2, \pi_1 \rangle; ev))$$

Un calcul direct utilisant la structure cartésienne fermée et les lois de naturalité et de cohérence de la force de T permet alors de montrer ' σ' '; $f_{A,B}^T = 'T(\sigma)'$, comme requis. La naturalité de $f_{A,B}^T$ s'obtient elle aussi par calcul direct. \square

Remarque. Le lemme ci-dessus peut être résumé par l'énoncé troublant : «tout foncteur ouvert est fermé».

Corollaire 7.5.13. *Soit $T : \mathbf{Inn} \rightarrow \mathbf{Inn}$ un foncteur ouvert. Par le Lemme 7.5.10 et la structure complète de l'ordre \leq , on peut définir :*

$$T^\omega = \bigsqcup_{i=0}^{\infty} T^i(1)$$

T^ω est alors un invariant minimal pour le foncteur T .

Démonstration. Les lemmes ci-dessus assurent que T vérifie les hypothèses du théorème de McCusker : il est *fermé*, continu, et préserve les injections et projections. \square

Chapitre 8

Modèle des points fixes

Sommaire

8.1 Jeux à gain	220
8.1.1 Décomposition en fils	220
8.1.2 Jeux à gain	221
8.1.3 La catégorie Gain	223
8.1.4 Relations avec la Partie I	225
8.2 Points fixes	227
8.2.1 Foncteurs ouverts à gain	227
8.2.2 Types forts pour le gain	228
8.2.3 Construction de boucle	229
8.2.4 Une catégorie μ -fermée	233
8.3 Sur la complétude	236

Dans ce chapitre, on va achever la construction du modèle de μLJ par la construction d'une catégorie μ -fermée concrète de jeux et stratégies totales. Pour ce faire, on utilise deux ingrédients. Le premier est la structure de types pseudo-polynomiaux forts sur **Par**, que l'on vient d'introduire. Le second est une technique standard en sémantique de jeux pour construire des catégories de stratégies totales : le *gain*. C'est un outil fréquent en théorie des jeux, il est par exemple fondamental dans l'étude du μ -calcul par des jeux de parité [AN01]. À la connaissance de l'auteur, la première utilisation d'une notion de gain en sémantique de jeux remonte au modèle d'Abramsky et Jagadeesan de MLL avec MIX [AJ94]. On peut par ailleurs en trouver une description introductive dans [Hy197]. L'idée est simple : on équipe chaque jeu d'une notion de gain, qui donne pour chaque partie un *gagnant* et un *perdant* parmi Joueur et Opposant. Le cas important est celui des parties infinies, puisque comme on l'a vu dans la première partie ce sont elles qui sont responsables des entorses à la totalité générées par composition. Pour chaque partie infinie, il devra en effet y avoir un perdant : l'absurdité apparaîtra alors du fait qu'on aura supposé nos stratégies *gagnantes* – c'est-à-dire qu'elle ne contiendront que des parties gagnantes.

Dans la première section, on va définir une catégorie de jeux **Gain** d'arènes à gain et de stratégies gagnantes. On montrera que parmi ces stratégies gagnantes, les totales sont stables par composition. À ce point, on fera le parallèle avec les notions étudiées dans la Partie I : par exemple, quelles sont les relations entre les conditions qu'on y donne sur les stratégies (finies, bornées, noéthériennes) et les conditions de gain sur **Gain**. Dans la seconde

section, on avancera dans la construction du modèle proprement dit en montrant que cette catégorie **Gain** hérite de **Par** (dont elle peut être vue comme une sous-catégorie) sa structure de types pseudo-polynomiaux forts. En particulier, la notion de foncteur ouvert sera adaptée dans **Gain** en une notion de *foncteur ouvert à gain*. On montrera ensuite que chaque foncteur ouvert à gain est automatiquement équipé d'un algèbre initiale et d'une coalgèbre terminale, rendant ainsi totale l'interprétation des formules de μLJ , et achevant la structure de catégorie μ -fermée.

8.1 Jeux à gain

Remarque. Dans les préliminaires, on a introduit la catégories **Par** sans imposer aux parties de condition de visibilité ou de parenthésage : elles étaient présentées à la place comme des propriétés sur les stratégies. Une autre présentation de la même catégorie de jeux, suivie par exemple dans [McC00], consiste à ne considérer que des parties légales visibles et bien parenthésées. Notons qu'il s'agit vraiment d'une présentation alternative de la même catégorie – ses objets sont les arènes et ses morphismes les stratégies innocentes et bien parenthésées. On peut donc sans rien changer à la catégorie ambiante restreindre les parties légales à celles qui sont visibles et bien parenthésées. C'est ce qu'on va faire dans ce dernier chapitre, dans la mesure où ça permet de simplifier certaines des constructions présentées.

8.1.1 Décomposition en fils

En préliminaire à la définition de jeux à gain, on va montrer comment décomposer chaque partie légale sur une arène A en *fils*. En effet, les différents fils d'une partie sur A correspondent à des parties sur différentes copies indépendantes : on demandera que joueur, pour gagner sur une telle partie doive gagner indépendamment sur chaque copie. Ce choix découle naturellement de la définition du gain sur les exponentielles dans [Hy197], où on demande qu'une partie sur $!A$ soit gagnante si et seulement sa restriction sur chaque copie est gagnante.

Définition 8.1.1. Soit A une arène. On note \mathcal{L}_A^ω l'ensemble des parties légales de A éventuellement infinies, on note \mathcal{L}_A^∞ l'ensemble des parties légales infinies de A . On utilise les mêmes notations Th_A , Th_A^ω et Th_A^∞ pour décrire les fils sur A , c'est-à-dire les parties légales avec un seul coup initial.

Définition 8.1.2 (Restriction sur un coup initial). Soit $s \in \mathcal{L}_A^\omega$, et i tel que s_i est un coup initial sur A . On définit alors la restriction de s à i comme la sous-partie de s contenant les coups héréditairement justifiés par s_i , avec les mêmes pointeurs.

Définition 8.1.3 (Décomposition en fils). Soit $s \in \mathcal{L}_A^\omega$. On définit la décomposition en fils de s , notés $\llbracket s \rrbracket$, par :

$$\llbracket s \rrbracket = \{s_{\uparrow i} \mid s_i \text{ est un coup initial}\}$$

Montrons maintenant quelques propriétés de cette décomposition en fils.

Lemme 8.1.1. Soit $s \in \mathcal{L}_A^\omega$, alors :

- s est de longueur paire si et seulement si quel que soit $t \in \llbracket s \rrbracket$, t est de longueur paire ;
- s est infinie si et seulement si quel que soit $t \in \llbracket s \rrbracket$, t est soit infini, soit de longueur paire.

Démonstration. Remarquons que Joueur joue toujours dans le fil du dernier coup d'Opposant. En effet, par visibilité Joueur doit pointer dans sa P -vue, or sa P -vue est incluse dans le fil du dernier coup Opposant. Cette remarque permet de prouver le premier énoncé par induction immédiate sur s . Si s est infinie, $\llbracket s \rrbracket$ contient des fils finis, et des fils infinis. Il existe alors un préfixe fini $s' \sqsubseteq^P s$ de longueur paire tel que tous les fils finis de s soient déjà complets dans s' . Par hypothèse, s' est de longueur paire donc tous ces fils sont également de longueur paire. \square

8.1.2 Jeux à gain

Dans cette sous-section, on va donner les définitions de base des jeux à gain, jusqu'à la construction de la catégorie **Gain**.

Définition 8.1.4. *Soit A une arène. Une fonction de gain sur A est une fonction*

$$G : \mathcal{L}_A^\omega \rightarrow \{W, L\}$$

déterminant si une partie s est un gain pour Joueur, et vérifiant les conditions suivantes :

- (1) $\forall s \in \mathcal{L}_A^\omega, G(s) = W \Leftrightarrow \forall t \in \llbracket s \rrbracket, G(t) = W$;
- (2) Si s est finie et s est de longueur paire, alors $G(s) = W$;
- (3) Si s est finie et s est de longueur impaire, alors $G(s) = L$.

Un jeu à gain \mathcal{A} est alors un couple $(A, G_{\mathcal{A}})$ où A est une arène, et $G_{\mathcal{A}}$ une fonction de gain sur A .

Notons que de par les relations entre une partie s et ses fils, une fonction de gain est entièrement définie et caractérisée par sa valeur sur les fils infinis. Il suffira donc par la suite de définir les fonctions de gains sur les fils infinis.

Lemme 8.1.2. *Soit A une arène, et considérons Th_A^∞ l'ensemble des fils infinis sur A . Soit $G : Th_A^\infty \rightarrow \{W, L\}$ n'importe quelle fonction. Il existe alors une unique fonction de gain G_A coïncidant avec G sur Th_A^∞ .*

Démonstration. On définit G_A d'abord sur l'ensemble des fils finis, par $G_A(t) = W$ si $|t| \equiv_2 0$ et $G_A(t) = L$ si $|t| \not\equiv_2 0$. Puis, on étend G_A à \mathcal{L}_A^ω par $G_A(s) = W$ si et seulement si quel que soit $t \in \llbracket s \rrbracket, G_A(t) = W$. Vérifions alors que G_A vérifie les trois conditions des fonctions de gain :

- La première condition est vérifiée par définition de l'image de G_A sur les parties à plusieurs fils ;
- Les deux conditions suivantes sont vérifiées par définition et par le Lemme 8.1.1. En effet, si s est finie et de longueur paire, c'est que tous ses fils sont de longueur paire, auquel cas on a bien $G_A(s) = W$.
- De même, si s est de longueur impaire, au moins l'un de ses fils est de longueur impaire et perdant, auquel cas s est également perdante.

G_A est donc une fonction de gain bien définie. C'est également la seule extension possible : en effet, sa valeur est forcée sur les parties à plusieurs fils par la condition (1), et sur les fils finis par la condition (2). \square

On va maintenant montrer comment étendre les constructions usuelles d'arènes présentées dans les préliminaires aux jeux à gain.

Lemme 8.1.3 (Unités). *Il n'existe qu'une seule fonction de gain sur les arènes 0 et 1, respectivement notée G_0 et G_1 . On note alors $\mathbf{0} = (0, G_0)$ et $\mathbf{1} = (1, G_1)$.*

Démonstration. L'unicité de la fonction de gain sur 0 et 1 est triviale par le Lemme 8.1.2. En effet, les parties légales sur ces arènes ne comportent pas de fil infini. \square

Lemme 8.1.4. *Soient A et B deux arènes. On a :*

$$Th_{A \times B}^\infty \simeq Th_A^\infty + Th_B^\infty$$

Démonstration. Immédiat, puisqu'un coup initial sur $A \times B$ est soit un coup initial sur A , soit un coup initial sur B . \square

Définition 8.1.5. *Le Lemme 8.1.4 permet de définir $G_{A \times B}$ en utilisant le Lemme 8.1.2 comme l'unique fonction de gain coïncidant avec G_A sur Th_A^∞ et G_B sur Th_B^∞ . On note $\mathcal{A} \times \mathcal{B} = (A \times B, G_{A \times B})$.*

Lemme 8.1.5. *Soient A et B deux arènes, on a à isomorphisme près :*

$$Th_{A+B}^\infty \simeq Th_A^\infty + Th_B^\infty$$

Démonstration. Élaborons un isomorphisme entre ces deux ensembles. Soit $t \in Th_{A+B}^\infty$. Par bon parenthésage, on a une seule réponse dans t à la question initiale q , qui est soit L , soit R : selon cette réponse, le reste de la partie sera soit entièrement inclu dans A , soit entièrement inclus dans B . On aura donc soit $t \upharpoonright_A = \epsilon$, soit $t \upharpoonright_B = \epsilon$. Notons qu'on ne peut pas avoir les deux, puisque t est supposé infini. On obtient donc à partir de t un élément de $Th_A^\infty + Th_B^\infty$. Dans l'autre sens, si $t \in Th_A^\infty + Th_B^\infty$ est de la forme $in_1(t_A)$, alors $qLt_A \in Th_{A+B}^\infty$. De même, si t est de la forme $in_2(t_B)$, $qRt_B \in Th_{A+B}^\infty$. Il est trivial de voir que ces deux transformations sont inverses. \square

Remarque. On a la chaîne d'isomorphismes déconcertante suivante :

$$\begin{aligned} Th_{A \times B}^\infty &\simeq Th_A^\infty + Th_B^\infty \\ &\simeq Th_{A+B}^\infty \end{aligned}$$

À isomorphisme près, $A \times B$ et $A + B$ ont les mêmes fils infinis. Cela peut paraître contre-intuitif, mais on peut préciser ça par le fait que pour $A \times B$, c'est Opposant qui choisit dans quelle composante on va jouer, alors que pour $A + B$ c'est le rôle de Joueur.

Définition 8.1.6. *Soient \mathcal{A} et \mathcal{B} deux jeux à gain. Alors, le Lemme 8.1.5 permet de définir $G_{\mathcal{A}+\mathcal{B}}$ en utilisant le Lemme 8.1.2 comme l'unique fonction de gain coïncidant avec $G_{\mathcal{A}}$ sur $Th_{\mathcal{A}}^\infty$ et avec $G_{\mathcal{B}}$ sur $Th_{\mathcal{B}}^\infty$. On note $\mathcal{A} + \mathcal{B} = (A + B, G_{\mathcal{A}+\mathcal{B}})$.*

Définition 8.1.7. *Soient \mathcal{A} et \mathcal{B} deux jeux à gain. Avec l'aide du Lemme 8.1.2, on définit $G_{\mathcal{A} \Rightarrow \mathcal{B}}$ comme l'unique fonction de gain sur $A \Rightarrow B$ ayant comme valeur sur les fils infinis :*

$$G_{\mathcal{A} \Rightarrow \mathcal{B}}(t) = W \Leftrightarrow (G_{\mathcal{A}}(t \upharpoonright_A) = W \Rightarrow G_{\mathcal{B}}(t \upharpoonright_B) = W)$$

Lemme 8.1.6. *Soit $s \in \mathcal{L}_{A \Rightarrow B}^\infty$, où A et B sont les arènes sous-jacentes de jeux à gain \mathcal{A} et \mathcal{B} , supposons également que s est gagnante pour $G_{\mathcal{A} \Rightarrow \mathcal{B}}$. Alors si $G_{\mathcal{A}}(s \upharpoonright_A) = W$, on a $G_{\mathcal{B}}(s \upharpoonright_B) = W$.*

Démonstration. Soit $t \in \llbracket s \upharpoonright_B \rrbracket$, il existe un coup initial i tel que $t = (s \upharpoonright_B) \upharpoonright_i$. Il existe donc i' tel que ce coup initial $(s \upharpoonright_B) \upharpoonright_i$ correspond au coup $s \upharpoonright_{i'}$, par conséquent on a l'égalité $(s \upharpoonright_B) \upharpoonright_i = (s \upharpoonright_{i'}) \upharpoonright_B$. Alors,

$$\llbracket (s \upharpoonright_{i'}) \upharpoonright_A \rrbracket \subset \llbracket s \upharpoonright_A \rrbracket \subset G_{\mathcal{A}}^{-1}(\{W\})$$

La dernière inclusion est vraie car s est supposée gagnante sur \mathcal{A} . Or, $s \upharpoonright_{i'}$ est gagnante car c'est un fil de s donc par définition du gain sur $\mathcal{A} \Rightarrow \mathcal{B}$, $(s \upharpoonright_{i'}) \upharpoonright_B = (s \upharpoonright_B) \upharpoonright_i = t$ est gagnante. \square

8.1.3 La catégorie Gain

Comme on s'en doute, les objets de la catégorie **Gain** seront les jeux à gain. Il reste, par contre, à définir les stratégies et à montrer les différents principes de leur composition.

Définition 8.1.8. *Soit A une arène et $\sigma : A$ une stratégie sur A . On définit l'ensemble des parties éventuellement infinies de σ par :*

$$\sigma^\omega = \{s \in \mathcal{L}_A^\omega \mid \forall s' \sqsubseteq^P s, s' \in \sigma\}$$

Définition 8.1.9. *Soit \mathcal{A} un jeu à gain. Une stratégie $\sigma : A$ est dite gagnante sur \mathcal{A} si et seulement si pour tout $s \in \sigma^\omega$, $G_{\mathcal{A}}(s) = W$.*

Proposition 8.1.7. *Soient \mathcal{A}, \mathcal{B} et \mathcal{C} des jeux à gain. Si $\sigma : \mathcal{A} \Rightarrow \mathcal{B}$ et $\tau : \mathcal{B} \Rightarrow \mathcal{C}$ sont deux stratégies gagnantes, alors $\sigma; \tau : \mathcal{A} \Rightarrow \mathcal{C}$ est gagnante.*

Démonstration. Par commodité, on va s'intéresser aux fils t de $\sigma; \tau$. Soit $t \in (\sigma; \tau)^\omega$. Si t est fini, t est également de longueur paire et $G_{\mathcal{A} \Rightarrow \mathcal{C}}(t) = W$. Si t est infini, on a pour chaque préfixe t' pair de t un témoin $u \in \sigma \parallel \tau$, or ces témoins forment eux aussi une chaîne de préfixes par déterminisme de σ et τ . Par conséquent, on a un témoin infini $u \in \sigma \parallel \tau$ pour t . Par définition de $G_{\mathcal{A} \Rightarrow \mathcal{C}}$, il faut montrer que si $G_{\mathcal{A}}(t \upharpoonright_A) = W$, alors $G_{\mathcal{C}}(t \upharpoonright_C) = W$. Supposons donc que $G_{\mathcal{A}}(t \upharpoonright_A) = W$, ou autrement dit que $G_{\mathcal{A}}(u \upharpoonright_A) = W$. Mais σ est gagnante, donc d'après le Lemme 8.1.6, on a également $G_{\mathcal{B}}(u \upharpoonright_B) = W$. Et encore d'après le Lemme 8.1.6 et le fait que τ est gagnante, on a $G_{\mathcal{C}}(u \upharpoonright_C) = W$, c'est-à-dire $G_{\mathcal{C}}(t \upharpoonright_C) = W$, et c'est fini. \square

Proposition 8.1.8. *Soient \mathcal{A}, \mathcal{B} et \mathcal{C} des jeux à gain. Si $\sigma : \mathcal{A} \Rightarrow \mathcal{B}$ et $\tau : \mathcal{B} \Rightarrow \mathcal{C}$ sont deux stratégies gagnantes et totales, alors $\sigma; \tau : \mathcal{A} \Rightarrow \mathcal{C}$ est totale.*

Démonstration. Comme on l'a vu dans la Partie I, la partialité de $\sigma; \tau$ implique l'existence d'une interaction $u \in \sigma \parallel \tau$ finie dans A et C , et infinie dans B . Chaque restriction $u \upharpoonright_A, u \upharpoonright_B$ et $u \upharpoonright_C$ est soit gagnante, soit perdante. Une analyse directe montre qu'on a alors nécessairement $u \upharpoonright_A$ de longueur paire et $u \upharpoonright_C$ de longueur impaire, donc $G_{\mathcal{A}}(u \upharpoonright_A) = W$ et $G_{\mathcal{C}}(u \upharpoonright_C) = L$. On utilise alors le fait que σ est gagnante, donc $G_{\mathcal{A} \Rightarrow \mathcal{B}}(u \upharpoonright_{\mathcal{A} \Rightarrow \mathcal{B}}) = W$, donc d'après le Lemme 8.1.6, comme $G_{\mathcal{A}}(u \upharpoonright_A) = W$, on a également $G_{\mathcal{B}}(u \upharpoonright_B) = W$. Mais τ est également gagnante, donc $G_{\mathcal{B} \Rightarrow \mathcal{C}}(u \upharpoonright_{\mathcal{B} \Rightarrow \mathcal{C}}) = W$, donc d'après le Lemme 8.1.6, on a $G_{\mathcal{C}}(u \upharpoonright_C) = W$, ce qui est absurde. Donc il ne peut pas exister de telle interaction infinie, donc $\sigma; \tau$ est totale. \square

Proposition 8.1.9. *Soit \mathcal{A} un jeu à gain. Alors on a $id_{\mathcal{A}} : \mathcal{A} \Rightarrow \mathcal{A}$, c'est-à-dire que l'identité est gagnante.*

Démonstration. Notons A_l l'occurrence de gauche de A , et A_r l'occurrence de droite. Soit t un fil de id_A , éventuellement infini. Supposons que $G_{\mathcal{A}}(t_{\downarrow A_l}) = W$. Mais par définition de l'identité, on a $t_{\downarrow A_l} = t_{\downarrow A_r}$, donc on a également $G_{\mathcal{A}}(t_{\downarrow A_r}) = W$ et t est une partie gagnante sur $\mathcal{A} \Rightarrow \mathcal{A}$. \square

Définition 8.1.10 (Catégorie **Gain**). *Les objets de **Gain** sont les jeux à gain et les morphismes sont les stratégies innocentes, bien parenthésées, totales et gagnantes.*

Notons qu'on a montré au dessus que cette classe de stratégies était stable par composition, et qu'elle contenait les identités. La composition est bien sûr associative, puisque c'est la composition usuelle entre stratégies. Montrons maintenant que **Gain** conserve la structure de **Inn** et **Par**, c'est-à-dire que tous les morphismes fondamentaux sont gagnants.

Proposition 8.1.10. *La catégorie **Gain** est cartésienne fermée, et équipée de coproduits faibles fonctoriels.*

Démonstration. Il suffit de montrer que les morphismes fondamentaux de cette structure sont gagnants, puisqu'ils vérifieront automatiquement les mêmes équations. Passons en revue les différentes structures de **Gain** :

- *Produits.* Les projections $\pi_1 : \mathcal{A} \times \mathcal{B} \rightarrow \mathcal{A}$ et $\pi_2 : \mathcal{A} \times \mathcal{B} \rightarrow \mathcal{B}$ sont gagnantes. C'est évident étant donnée la définition du gain sur $\mathcal{A} \Rightarrow \mathcal{B}$ et $\mathcal{A} \times \mathcal{B}$. De même, si $\sigma : \mathcal{C} \rightarrow \mathcal{A}$ et $\tau : \mathcal{C} \rightarrow \mathcal{B}$ sont gagnantes, $\langle \sigma, \tau \rangle$ est gagnante. En effet, un fil de $\langle \sigma, \tau \rangle$ est soit un fil de σ soit un fil de τ , dans tous les cas gagnant si σ et τ sont gagnantes. Donc **Gain** est cartésienne.
- *Coproduits.* Les injections $in_1 : \mathcal{A} \rightarrow \mathcal{A} + \mathcal{B}$ et $in_2 : \mathcal{B} \rightarrow \mathcal{A} + \mathcal{B}$ sont également gagnantes. En effet, un fil de in_1 est de la forme qLs , où (au pointeur des coups initiaux à gauche près) $s \in id_A$. Or, id_A est gagnante. Donc, d'après la définition du gain sur $\mathcal{A} + \mathcal{B}$ et le Lemme 8.1.6, comme on a $G_{\mathcal{A}}(qLs_{\downarrow A_l}) = G_{\mathcal{A}}(s_{\downarrow A_l}) = W$, on a également $G_{\mathcal{A}}(s_{\downarrow A_r}) = G_{\mathcal{A}}(qLs_{\downarrow A_r}) = W$. La preuve est la même pour in_2 . Supposons maintenant que $\sigma : \mathcal{A} \rightarrow \mathcal{C}$ et $\tau : \mathcal{B} \rightarrow \mathcal{C}$ sont gagnantes, et soit $t \in [\sigma|\tau]$. Supposons $t_{\downarrow \mathcal{A}+\mathcal{B}}$ gagnante. Par définition de $[\sigma|\tau]$, t est nécessairement de la forme (à réordonnement de pointeurs près) $qCq+Lt'$ avec qCt' un fil de σ , ou $qCq+Rt'$ avec qCt' un fil de τ . Donc l'hypothèse $t_{\downarrow \mathcal{A}+\mathcal{B}}$ gagnante de vient, par définition du gain sur $\mathcal{A} + \mathcal{B}$, soit $(qCt')_{\downarrow \mathcal{A}}$ gagnante, soit $(qCt')_{\downarrow \mathcal{B}}$ gagnante. On applique alors respectivement le gain de σ ou τ , d'où on déduit que $(qCt')_{\downarrow \mathcal{C}} = t_{\downarrow \mathcal{C}}$ est gagnante, ce qui conclut.
- *Fermeture.* Remarquons d'abord que si $\sigma : \mathcal{A} \times \mathcal{B} \rightarrow \mathcal{C}$, σ est gagnante si et seulement si $\Lambda(\sigma) : \mathcal{B} \rightarrow \mathcal{A} \Rightarrow \mathcal{C}$ est gagnante : il est évident que par définition du gain sur \times et \Rightarrow , les conditions de gain coïncident. Donc puisque $\Lambda(ev) = id_{\mathcal{A} \Rightarrow \mathcal{B}}$, on en déduit que ev est gagnante. Comme ev et Λ satisfont les mêmes équations que dans **Inn**, **Gain** est cartésienne fermée. \square

Notons qu'il est direct de généraliser la somme binaire introduite ci-dessus à une somme d'arité quelconque. Cette remarque aura son importance pour généraliser le théorème de factorisation aux jeux à gain, puisque la preuve utilise le caractère gagnant des injections canoniques des sommes n -aires.

8.1.4 Relations avec la Partie I

On a donc à ce point un modèle de la logique intuitionniste avec sommes, mais c'était déjà le cas de la catégorie **Par**. Surtout, c'est un modèle où les stratégies restent *totales*, ce qui d'une part implique que LJ est cohérente (voir le Corollaire 8.2.15 pour plus de détails sur ce raisonnement), mais permet surtout de se rapprocher de la complétude. Par ailleurs, on avait construit dans la Partie I de nombreuses catégories semblables de jeux et stratégies totales, permettant d'interpréter LJ . Il est alors naturel de s'interroger sur les relations entre ces différentes approches : est-ce qu'elles définissent la même interprétation de la logique ? On va montrer ici que la réponse est oui. Hyland décrit dans [Hyl97] un *principe de justice* consistant à imposer aux fonctions de gain qu'elles fassent perdre tout joueur qui, dans une partie infinie, produirait une fourche consciente infinie. C'est cette notion de *principe de justice* qui fera le pont entre les deux approches.

Définition 8.1.11. *Soit \mathcal{A} un jeu à gain. On dit que \mathcal{A} satisfait le principe de justice si et seulement si tout joueur causant une fourche consciente infinie est perdant pour \mathcal{A} . Autrement dit, on a les deux conditions :*

- Soit s une partie, alors si Joueur a une fourche consciente infinie dans s , il doit perdre. Autrement dit, $G_{\mathcal{A}}(s) = L$.
- Soit s un fil, alors si Opposant a une fourche consciente infinie dans s , il doit perdre. Autrement dit, $G_{\mathcal{A}}(s) = W$.

Lemme 8.1.11. *Soit \mathcal{A} un jeu à gain tel que l'arène A a une profondeur bien fondée. Alors \mathcal{A} satisfait le principe de justice si et seulement si l'énoncé suivant est vérifié : pour toute partie infinie s sur A , $G_{\mathcal{A}}(s) = W$ si et seulement si Joueur n'a pas de fourche consciente infinie dans s .*

Démonstration. Ce résultat pourrait paraître trivial, mais il faut en fait utiliser plusieurs corollaires du théorème de finitude forte pour le démontrer. De plus, il est faux si on relâche l'hypothèse de bonne fondation pour A . Montrons d'abord la première implication, et supposons que \mathcal{A} satisfait le principe de justice. Alors, supposons que $G_{\mathcal{A}}(s) = W$. Clairement, Joueur ne peut pas avoir de fourche consciente infinie sur s , sinon on aurait $G_{\mathcal{A}}(s) = L$ par le principe de justice. Supposons maintenant que Joueur n'a pas de fourche consciente infinie sur s . s est cependant infinie, et A a une profondeur bien fondée, donc Opposant doit avoir une fourche consciente infinie d'après le Corollaire 4.2.18, d'où $G_{\mathcal{A}}(s) = W$ par définition du principe de justice.

Montrons la seconde implication, et supposons l'énoncé ci-dessus satisfait. Pour montrer que \mathcal{A} satisfait le principe de justice, deux énoncés à montrer :

- Supposons que Joueur ait une fourche consciente infinie dans s . Par l'énoncé supposé, on a immédiatement $G_{\mathcal{A}}(s) = L$.
- Supposons que Opposant ait une fourche consciente infinie dans s , on doit montrer alors que $G_{\mathcal{A}}(s) = W$, c'est à dire que Joueur n'a pas de fourche consciente infinie dans s . Mais puisque A est à profondeur bien fondée, c'est une conséquence du Lemme 4.3.2 : il est impossible que les deux joueurs aient tous deux une fourche consciente infinie, donc Joueur n'en a pas, donc $G_{\mathcal{A}}(s) = W$.

□

Lemme 8.1.12. *Soit \mathcal{A} un jeu à gain satisfaisant le principe de justice, sur une arène A bien fondée. Alors les stratégies gagnantes sur \mathcal{A} sont exactement les stratégies naïthériennes sur A .*

Démonstration. Rappelons d'abord que sur dans une arène bien fondée, les stratégies noethériennes sont exactement celles qui n'ont pas de fourche consciente infinie. La preuve est alors triviale par le Lemme 8.1.11. \square

Notons en passant que si on omet l'hypothèse de bonne fondation de A , on garde un des sens de l'équivalence ci-dessus.

Lemme 8.1.13. *Soit A un jeu à gain satisfaisant le principe de justice. Alors si $\sigma : A$ est noethérienne et innocente, elle est également gagnante sur A .*

Démonstration. Soit $s \in \sigma^\omega$. Si s est finie, s est évidemment gagnante car de longueur paire. Sinon s est infinie, mais s est tout de même à profondeur bien fondée et sans fourche consciente de Joueur, puisque σ est noethérienne. Donc par le Corollaire 4.2.18, s contient une fourche consciente infinie d'Opposant et $G_{\mathcal{A}}(s) = W$. \square

Lemme 8.1.14. *Soit A une arène bien fondée, il existe un unique jeu à gain \mathcal{A} d'arène sous-jacente A satisfaisant le principe de justice.*

Démonstration. La fonction de gain $G_{\mathcal{A}}$ est définie comme suit :

- Si s a plusieurs fils, alors $G_{\mathcal{A}}(s) = W \Leftrightarrow \forall t \in \llbracket s \rrbracket, G_{\mathcal{A}}(t) = W$;
- Si t est un fil fini, alors $G_{\mathcal{A}}(t) = W$ si et seulement si t est de longueur paire ;
- Si t est un fil infini, alors $G_{\mathcal{A}}(t) = W$ si et seulement si Joueur n'a pas de fourche consciente infinie dans φ_t .

Il est évident que c'est une fonction de gain. Elle satisfait immédiatement le principe de justice, puisque par définition elle satisfait la caractérisation du Lemme 8.1.11. C'est également la seule fonction de gain respectant le principe de justice : les deux premières conditions doivent nécessairement être satisfaites puisque c'est une fonction de gain, et la troisième car elle doit satisfaire le principe de justice. Or, ces trois conditions caractérisent $G_{\mathcal{A}}$. \square

Notons que si une arène A n'a pas une profondeur bien fondée, il n'existe pas nécessairement de fonction de gain sur A satisfaisant le principe de justice. En effet, on a vu en Figure 4.5 qu'on pouvait avoir dans une même partie une fourche consciente infinie pour chaque joueur. Mais dans ce cas, quel gagnant choisir ? Le principe de justice impose que Joueur soit gagnant puisque Opposant a une fourche consciente infinie, mais il impose aussi que Joueur soit perdant, puisqu'il a également une fourche consciente infinie.

Proposition 8.1.15. *La catégorie **Noeth** des arènes bien fondées et stratégies noethériennes et totales est isomorphe à la catégorie **Just** des jeux à gain satisfaisant le principe de justice sur des arènes bien fondées et des stratégies gagnantes et totales.*

Démonstration. On construit $F : \mathbf{Noeth} \rightarrow \mathbf{Just}$ par :

- Sur les arènes, $F(A) = \mathcal{A}$ où \mathcal{A} est l'unique jeu à gain donné par le Lemme 8.1.14.
- Sur les stratégies, $F(\sigma) = \sigma$. Par le Lemme 8.1.15, σ est bien gagnante sur $F(A)$.

On construit $G : \mathbf{Just} \rightarrow \mathbf{Noeth}$ par :

- Sur les jeux, $G(A, G) = A$;
- Sur les stratégies, $G(\sigma) = \sigma$. Par le Lemme 8.1.15, σ est en effet noethérienne.

Il est clair que $GF = Id$. Pour montrer que $FG = Id$ sur les objets, il suffit d'utiliser le Lemme 8.1.14 : la fonction de gain G recomposée est forcément identique à la première, puisque celle-ci est unique. \square

Maintenant, notons que les unités sont représentées dans tous les cas par des arènes bien fondées. Si les éventuels atomes sont interprétés par des arènes bien fondées, ce sera préservé par l'interprétation des constructeurs de formules, et toutes les arènes nécessaires pour interpréter LJ seront bien fondées. Ainsi, on obtient en fait exactement le même modèle de la logique, qu'on passe par des conditions internes sur les stratégies (stratégies noethériennes) ou par une définition externe de gain. Pourquoi alors passer par le gain, qui demande plus de travail maintenant que les stratégies noethériennes sont bien définies? La réponse est que si les arènes finies, *a fortiori* bien fondées suffisent à interpréter une logique finitaire, elles ne suffisent plus pour μLJ . En effet, les arènes utilisées ne seront plus bien fondées à cause de la présence de cycles. Par conséquent certaines stratégies gagnantes parfaitement raisonnables ne seront plus noethériennes : un exemple emblématique est celui de la stratégie identité sur les entiers paresseux.

Réalisabilité. On peut également relever un parallèle précis entre cette notion de gain propre à la sémantique de jeux et la technique de réalisabilité, fréquemment utilisée en théorie de la preuve : on peut par exemple la voir à l'œuvre dans cette thèse, dans la première partie, pour montrer la normalisation du λ -calcul unaire. Dans un cas, on définit un prédicat de gain sur les parties, dans l'autre un prédicat de réalisabilité sur les termes/stratégies, mais dans les deux cas cette définition se fait par induction sur les formules. Puis, on montre par induction sur la structure des termes que les termes typés/les preuves s'interprètent comme des stratégies gagnantes ou comme des réalisateurs.

8.2 Points fixes

Maintenant qu'on a décrit la catégorie servant de canevas au modèle des points fixes, passons à la construction de la catégorie μ -fermée proprement dite. La première chose à faire est de montrer que la structure de types pseudo-polynomiaux forts de **Par** se transporte sur **Gain**. Il faut donc enrichir la notion de foncteur ouvert par le gain, montrer que les forces sont des stratégies gagnantes, et qu'elles gardent les mêmes conditions de naturalité : ce sera l'objet de la première sous-section. Puis, il faudra montrer que ces *foncteurs ouverts à gain* admettent des algèbres initiales et des coalgèbres terminales, ce qui conclura la construction de la catégorie μ -fermée : on aura donc un modèle de μLJ . Finalement, on explorera dans une dernière sous-section le problème de la complétude de ce modèle.

8.2.1 Foncteurs ouverts à gain

Donnons une notion de foncteur ouvert pour **Gain**, et explorons ses propriétés. D'une part, les foncteurs ouverts à gain devront se comporter comme un foncteur ouvert sur la composante «arène» des jeux à gain. D'autre part, ils devront être forts. On connaît déjà leurs forces : ce sont celles obtenues par composition verticale dans le chapitre précédent. Il faudra donc assurer que ces forces sont bien gagnantes.

Définition 8.2.1. *Un foncteur ouvert à gain est un foncteur*

$$\mathcal{T} : (\mathbf{Gain}^{op} \times \mathbf{Gain})^n \rightarrow \mathbf{Gain}$$

dont l'action sur les arènes est donnée par un foncteur ouvert et qui est fort en chacun de ses arguments, c'est-à-dire que pour toute factorisation $(\mathbf{Gain}^{op} \times \mathbf{Gain})^n = \mathfrak{C} \times \mathfrak{D}$ où $\mathfrak{C} = \mathbf{Gain}$

ou $\mathfrak{C} = \mathbf{Gain}^{op}$, et où \mathfrak{D} est un produit de copies de \mathbf{Gain} et \mathbf{Gain}^{op} , pour tout objet D de \mathfrak{D} , la force $\theta_{\Gamma, A}^{T(-, D)}$ (ou $\rho_{\Gamma, A}^{T(-, D)}$, si $\mathfrak{C} = \mathbf{Gain}^{op}$) donnée par la structure de types pseudo-polynomiaux forts de \mathbf{Par} est gagnante pour tous Γ et A de \mathbf{Gain} .

Proposition 8.2.1. *La classe des foncteurs ouverts à gain comprend les foncteurs constants, l'identité, la somme, les produits et l'exponentiation $- \Rightarrow \mathcal{A}$ pour tout jeu à gain \mathcal{A} . De plus, elle est stable par composition et par contraction.*

Démonstration. Les forces et contraforces sont données par la structure de types pseudo-polynomiaux forts de \mathbf{Par} . Pour les constantes, l'identité, les sommes, les produits et l'exponentiation, ces forces sont données par la structure cartésienne fermée et les coproduits faibles fonctoriels donc sont toujours gagnantes. Les forces et contraforces pour la composée de deux foncteurs ouverts et la contraction sont égales (encore par la structure de types pseudo-polynomiaux forts) à des compositions de forces et contraforces déjà supposées gagnantes, donc toujours gagnantes. \square

Ainsi, on a une classe de foncteurs forts pour \mathbf{Gain} , assez grande pour donner une structure de types pseudo-polynomiaux forts. Étant donné que les forces sont égales à celles qu'on avait pour \mathbf{Par} , elles vérifient également les mêmes équations. Cependant, il faut tout de même vérifier que ces forces et contraforces sont toujours naturelles en leur paramètre foncteur : on sait que ces forces et contraforces sont naturelles en T dans \mathbf{Par} , donc qu'elles commutent avec toute transformation qui est naturelle entre foncteurs ouverts de \mathbf{Par} ; mais il y a potentiellement plus de transformations naturelles entre foncteurs ouverts dans \mathbf{Gain} que dans \mathbf{Par} . En effet, une transformation naturelle entre foncteurs ouverts à gain ne doit commuter qu'avec les stratégies gagnantes, alors qu'une transformation naturelle entre foncteurs ouverts dans \mathbf{Par} doit commuter avec toutes les stratégies innocentes et bien parenthésées. Cependant, on va voir dans la prochaine sous-section que ce n'est pas le cas : une transformation naturelle entre foncteurs ouverts à gain sera toujours une transformation naturelle entre les foncteurs ouverts sous-jacents, donc une transformation naturelle ouverte.

8.2.2 Types forts pour le gain

Pour obtenir la naturalité des forces et contraforces en leur paramètre fonctoriel, on va étendre le théorème de factorisation des transformations naturelles entre foncteurs ouverts aux transformations naturelles entre foncteurs ouverts à gain. Commençons par prendre deux foncteurs ouverts à gain \mathcal{S} et \mathcal{T} , et considérons une transformation naturelle $\eta : \mathcal{S} \xrightarrow{\bullet} \mathcal{T}$. En oubliant le gain, on obtient pour chaque jeu \mathcal{A} une stratégie $\eta_{\mathcal{A}} : SA \rightarrow TA$. On ne peut pas en tirer directement une transformation naturelle sur \mathbf{Par} , cependant la famille $\eta_{\mathcal{A}}$ commute avec toutes les stratégies gagnantes, ce qui donnera en pratique suffisamment de commutations pour appliquer le raisonnement de la Section 7.3.2. Certains lemmes sont nécessaires pour montrer qu'on a suffisamment de stratégies gagnantes, en particulier il faut montrer que tout jeu à gain peut être approximé par une chaîne croissante de jeux à gain sur des arènes finies, de sorte que les injections et projections soient gagnantes.

Lemme 8.2.2 (Dépliage). *Soit \mathcal{A} un jeu à gain. Le dépliage \tilde{A} de l'arène A peut alors être étendu en un jeu à gain \tilde{A} tel que l'isomorphisme $\gamma_{\mathcal{A}} : \mathcal{A} \rightarrow \tilde{\mathcal{A}}$ et son inverse sont gagnants.*

Démonstration. Puisque A et \tilde{A} ont les mêmes parties à isomorphisme près, on équipe \tilde{A} de la même fonction de gain que sur \mathcal{A} . Il est alors direct que $\gamma_{\mathcal{A}}$ et $\gamma_{\mathcal{A}}^{-1}$ sont gagnantes. \square

Lemme 8.2.3 (Approximation). *Soit \mathcal{F} un jeu à gain dont l'arène sous-jacente F est une forêt infinie. Alors il existe une famille $\mathcal{F}_1, \mathcal{F}_2, \dots$ de jeux à gains formant une chaîne strictement croissante sur les arènes sous-jacentes :*

$$F_1 \trianglelefteq F_2 \trianglelefteq F_3 \trianglelefteq \dots$$

Et telles que pour tout i , le morphisme de projection $\mathbf{proj}_i : F \rightarrow F_i$ est gagnant sur $\mathcal{F} \Rightarrow \mathcal{F}_i$.

Démonstration. Pour i un entier, on définit F_i comme la forêt préfixe de F de profondeur $2i + 1$. Par la condition d'alternance Opposant/Joueur sur les arènes, les derniers coups des branches de F_i sont Opposant. Par ailleurs, chaque partie F_i est équipé d'une fonction de gain $G_{\mathcal{F}_i}$ qui est simplement la restriction à $\mathcal{L}_{F_i}^\omega$ de $G_{\mathcal{F}}$. \mathbf{proj}_i est alors gagnante : chacune de ses parties est gagnante puisque c'est une partie de l'identité, et elle est totale car composée de parties de id_F tronquées après un coup Joueur. \square

Théorème 8.2.4 (Factorisation). *Soit $\eta : \mathcal{S} \xrightarrow{\bullet} \mathcal{T}$ une transformation naturelle entre foncteurs ouverts à gain. Alors pour tout jeu à gain \mathcal{A} , on a $\eta_{\mathcal{A}} = \eta[id_{\mathcal{A}}]$.*

Démonstration. Toutes les stratégies utilisées pour les Lemmes 7.3.10, 7.3.11 et 7.3.12 correspondent bien à des stratégies gagnantes. En fait, peu importe la fonction de gain utilisée : ces stratégies n'ont que des fils finis, et donc sont automatiquement gagnantes par définition des fonctions de gain. Les commutations utilisées pour les Lemmes 7.3.13, 7.3.14 et 7.3.15 sont toujours valides, par la structure cartésienne fermée avec coproduits faibles fonctoriels de **Gain**. On a donc la factorisation des transformations naturelles sur les jeux à gain dont l'arène sous-jacente est une forêt finie. Finalement, les deux lemmes ci-dessus permettent de passer aux forêts infinies, puis aux arènes quelconques. \square

Corollaire 8.2.5 (Types forts pour **Gain**). *La catégorie **Gain** a les types pseudo-polynomiaux forts.*

Démonstration. Il manquait juste la naturalité de $\theta_{\Gamma, \mathcal{A}}^T$ et de $\rho_{\Gamma, \mathcal{A}}^N$ en \mathcal{T} et \mathcal{N} . C'est une conséquence directe du théorème de factorisation. \square

8.2.3 Construction de boucle

Dans cette sous-section, on en vient finalement à un des points clé du modèle : la construction de boucles, tirant parti de la présence de cycles dans les arènes et permettant ainsi de donner une représentation finie aux points fixes, contrairement à l'expansion infinie de McCusker. On va introduire ici cette construction sur les arènes et l'enrichir par une condition de gain, et montrer qu'on obtient une algèbre initiale et une coalgèbre terminale pour chaque foncteur ouvert à gain covariant et unaire.

Définition 8.2.2 (Construction de boucle). *Soit T une arène ouverte. Pour chaque nom X telle que T est stricte en X (aucun trou \square_X n'est initial), on définit une nouvelle arène ouverte $\rho X.T$ comme suit :*

$$\begin{aligned} M_{\rho X.T} &= M_T \setminus \square_X \\ \lambda_{\rho X.T} &= \lambda_T \upharpoonright M_{\rho X.T} \\ m \vdash_{\rho X.T} n &\Leftrightarrow \bigvee \left\{ \begin{array}{l} m \vdash_T n \\ m \vdash_T \xi \cdot \square_X \wedge n \in I_T \end{array} \right. \\ I_{\rho X.T} &= I_T \end{aligned}$$

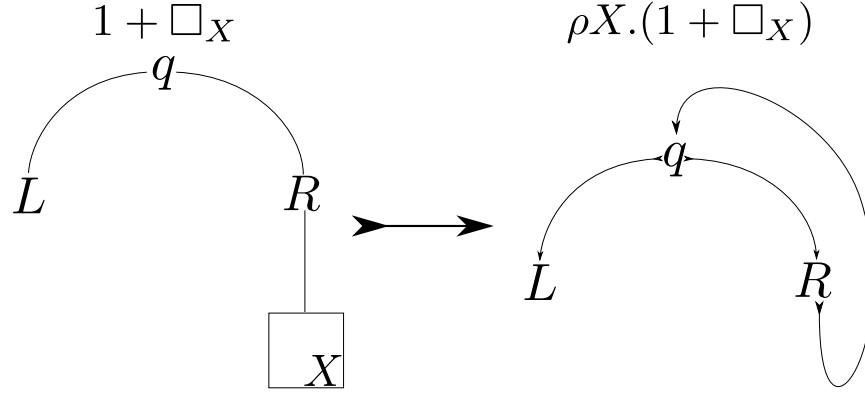


FIG. 8.1 – Construction de boucle.

Cette construction est illustrée dans la Figure 8.1. Notons que puisqu'on a supposé que \square_X n'apparaît jamais à la racine, il existe toujours un coup m qui le justifie, du moins dès que \square_X apparaît dans T . Cela assure que cette construction génère toujours un cycle dans l'arène.

Définition 8.2.3 (Profondeur de boucle). *Soit T une arène ouverte stricte et $s \in \mathcal{L}_{\rho X.T}^\omega$. Rappelons qu'un chemin de s est une suite $s_{\phi(1)}, s_{\phi(2)}, \dots$ éventuellement infinie, telle que pour tout i , $s_{\phi(i+1)}$ pointe vers $s_{\phi(i)}$ dans s . La profondeur de boucle de s , ou juste profondeur de s si le contexte est clair, est la borne supérieure (dans $\mathbb{N} \cup \{+\infty\}$) du nombre de fois que tout chemin $s_{\phi(1)}, s_{\phi(2)}, \dots$ de s passe par la boucle externe.*

Proposition 8.2.6. *Pour toute arène T stricte, il y a un isomorphisme de chemins ϕ (voir Section 2.4.2) entre l'arène $\rho X.T$ et l'arène T^ω obtenue à la Section 7.5.3 par itération infinie de T .*

Démonstration. On construit une bisimulation entre les chemins de $\rho X.T$ et T^ω : deux chemins sont en relation si le dernier coup du chemin dans T^ω est une copie du dernier coup du chemin dans $\rho X.T$. Par construction de T^ω et $\rho X.T$, les chemins ont des ensembles de successeurs isomorphes, et toujours en relation. \square

On notera que cette proposition permet d'avoir immédiatement le corollaire suivant :

Corollaire 8.2.7. *Soit T une arène ouverte stricte, alors l'arène $\rho X.T$ est un invariant minimal pour le foncteur ouvert T .*

Démonstration. On a vu que ces deux arènes sont reliées par un isomorphisme de chemins. Par le théorème de Laurent (Section 2.4.2), elles sont donc isomorphes dans **Par**. Par ailleurs, on sait par le développement de la Section 7.5.3 que T^ω est un invariant minimal pour T , donc c'est aussi le cas de $\rho X.T$. \square

Ainsi, cette construction de boucle permet de donner une représentation finie à l'invariant minimal pour T donné par le théorème de McCusker. Montrons maintenant comment l'enrichir avec du gain. Étant un isomorphisme de chemins, ϕ_T génère librement un isomorphisme de parties $\phi_T : \mathcal{L}_{\rho X.T}^\omega \rightarrow \mathcal{L}_{T^\omega}^\omega$ dont l'action commute avec le préfixe et avec la récupération du préfixe justificateur. À travers cet isomorphisme, on peut voir toute partie dans $\mathcal{L}_{\rho X.T}^\omega$ comme une partie dans $\mathcal{L}_{T^\omega}^\omega$ et réciproquement, et ce de façon unique.

Proposition 8.2.8. *Soit $s \in \mathcal{L}_{\rho X.T}^\omega$ de profondeur n . Alors $\phi(s) \in \mathcal{L}_{T^n(1)}^\omega$ mais $\phi(s) \notin \mathcal{L}_{T^{n-1}(1)}^\omega$.*

Démonstration. Par définition de la profondeur, c'est clair que $\phi(s) \in \mathcal{L}_{T^n(1)}^\omega$. Mais si on avait aussi $\phi(s) \in \mathcal{L}_{T^{n-1}(1)}^\omega$, alors on aurait également forcément s de profondeur strictement inférieure à n , ce qui est absurde. \square

Définition 8.2.4 (Plus petit point fixe). *Soit $\mathcal{T} : \mathbf{Gain} \rightarrow \mathbf{Gain}$ un foncteur ouvert à gain tel que l'arène ouverte T est stricte. On définit un jeu à gain $\mu X.T$ comme suit. Son arène sous-jacente est $\rho X.T$, et si $s \in \mathcal{L}_{\rho X.T}^\omega$, on définit sa fonction de gain G par :*

- Si s est de profondeur infinie, $G(s) = L$.
- Sinon, soit n la profondeur de s , on pose $G(s) = G_{T^n(1)}(\phi(s))$.

Définition 8.2.5 (Plus grand point fixe). *Soit $\mathcal{T} : \mathbf{Gain} \rightarrow \mathbf{Gain}$ un foncteur ouvert à gain tel que l'arène ouverte T est stricte. On définit un jeu à gain $\nu X.T$ comme suit. Son arène sous-jacente est $\rho X.T$, et si $s \in \mathcal{L}_{\rho X.T}^\omega$, on définit sa fonction de gain G par :*

- Si s est de profondeur infinie, $G(s) = W$.
- Sinon, soit n la profondeur de s , on pose $G(s) = G_{T^n(1)}(\phi(s))$.

Proposition 8.2.9. *Soit \mathcal{T} un foncteur ouvert à gain d'arène ouverte sous-jacente stricte. Alors il existe un isomorphisme de chemins évident entre $\rho X.T$ et $T(\rho X.T)$, qui génère un isomorphisme $\chi_T : T(\mu X.T) \rightarrow T$. Alors :*

- $\chi_T : T(\mu X.T) \rightarrow \mu X.T$ est une stratégie gagnante ;
- $\chi_T^{-1} : \nu X.T \rightarrow T(\nu X.T)$ est une stratégie gagnante.

Démonstration. La définition des conditions de gain sur $\mu X.T$ et $\nu X.T$ fait que les conditions de gain coïncident à gauche et à droite, donc χ_T et χ_T^{-1} sont bien gagnantes. \square

Théorème 8.2.10. *Soit \mathcal{T} un foncteur ouvert à gain d'arène ouverte sous-jacente stricte, alors $(\mu X.T, \chi_T)$ est une algèbre initiale pour \mathcal{T} .*

Démonstration. Soit (\mathcal{A}, σ) une autre algèbre de \mathcal{T} . Il faut vérifier qu'il existe une unique stratégie gagnante $\sigma^\dagger : \mu X.T \Rightarrow \mathcal{B}$ telle que le diagramme suivant commute.

$$\begin{array}{ccc} T(\mu X.T) & \xrightarrow{\mathcal{T}(\sigma^\dagger)} & T(\mathcal{B}) \\ i_T \downarrow & & \downarrow \sigma \\ \mu X.T & \xrightarrow{\sigma^\dagger} & \mathcal{B} \end{array}$$

Intuitivement on veut juste itérer σ :

$$\dots \xrightarrow{\mathcal{T}^3(\sigma)} \mathcal{T}^3(\mathcal{B}) \xrightarrow{\mathcal{T}^2(\sigma)} \mathcal{T}^2(\mathcal{B}) \xrightarrow{\mathcal{T}(\sigma)} \mathcal{T}(\mathcal{B}) \xrightarrow{\sigma} \mathcal{B}$$

Et prendre la limite. Malheureusement, aucune propriété de **Gain** n'assure directement que cette limite existe. On va cependant voir qu'on peut la construire en utilisant l'enrichissement en ordres partiels complets de **Par**. On commence par définir les itérées partielles de σ comme suit :

$$\begin{aligned} \sigma^{(1)} &= \sigma \\ \sigma^{(n+1)} &= \mathcal{T}^n(\sigma); \sigma^{(n)} \end{aligned}$$

On a pas en général $\sigma^{(n)} \subseteq \sigma^{(n+1)}$, ce qui empêche de conclure directement. Par contre, on a bien pour tout n :

$$\sigma_{|T^\omega \Rightarrow B}^{(n)} \subseteq \sigma_{|T^\omega \Rightarrow B}^{(n+1)}$$

Et toutes les stratégies de cette chaîne sont des stratégies (éventuellement partielles) sur $T^\omega \Rightarrow B$. On peut donc définir, par complétude de l'ordre sur les stratégies dans **Par** :

$$\sigma^{(\omega)} = \bigcup_{i \in \mathbb{N}_*} \sigma^{(i)}$$

C'est bien cette stratégie $\sigma^{(\omega)}$ qui nous intéresse, mais encore faut-il la rapporter à l'arène $\rho X.T \Rightarrow B$. Notons que l'isomorphisme de parties $\phi : \mathcal{L}_{\rho X.T}^\omega \rightarrow \mathcal{L}_{T^\omega}^\omega$ génère, en l'étendant par l'identité sur B , un isomorphisme de parties $\phi' : \mathcal{L}_{\rho X.T \Rightarrow B}^\omega \rightarrow \mathcal{L}_{T^\omega \Rightarrow B}^\omega$. On définit alors $\sigma^\dagger = \phi'^{-1}(\sigma^{(\omega)})$. Bien évidemment, σ^\dagger ainsi obtenue est innocente puisqu'elle est en correspondance directe avec $\sigma^{(\omega)}$, obtenue comme limite de stratégies innocentes.

La stratégie σ^\dagger est totale. Encore une fois, il suffit de le montrer pour $\sigma^{(\omega)}$. Supposons par l'absurde que $s \in \sigma^{(\omega)}$ et que sa n'ait pas de réponse. En particulier, il existe n tel que $s \in \sigma^{(n)}$. Alors, c'est que $\sigma^{(n)}$ n'a pas de réponse à sa dans $\rho X.T \Rightarrow B$. Mais on a également $s \in \sigma^{(n+1)}$, qui est totale puisque c'est le résultat d'une composition de stratégies totales. Or la réponse de $\sigma^{(n+1)}$ à a est nécessairement dans T^ω : la réponse de σ à un coup initial de B ne peut pas être directement dans B à gauche, à cause du fait que les trous ne peuvent pas apparaître à la racine des arènes ouvertes. Donc, il existe bien un coup b tel que $sab \in \sigma^{(n+1)}$ et $sab \in \mathcal{L}_{T^\omega \Rightarrow B}$, donc $sab \in \sigma^{(\omega)}$.

Montrons que σ^\dagger est gagnante. En effet, soit s éventuellement infinie dans σ^\dagger , telle que $s' = s|_{\rho X.T}$ est gagnante sur $\mu X.T$. Par définition du gain sur $\mu X.T$, cela signifie que s' a une profondeur finie n , et que $\phi(s')$ est gagnante sur $\mathcal{T}^n(\mathbf{1})$. Mais dans ce cas, on a en fait $\phi'(s) \in \sigma^{(n)}$, autrement s' serait nécessairement au moins à profondeur $n+1$. Or $\sigma^{(n)}$ est gagnante puisque c'est le résultat d'une composition de stratégies gagnantes, donc $\phi'(s)|_B = s|_B$ est bien gagnante sur B .

Finalement, il faut montrer qu'il n'y a pas d'autre stratégie $f : \mu X.T \Rightarrow B$ faisant commuter le diagramme ci-dessus. Supposons que ce soit le cas. On a alors $\mathcal{T}(f); \sigma = \chi_{\mathcal{T}}; f$ puisque f est un morphisme d'algèbres. Autrement dit :

$$f = \chi_{\mathcal{T}}^{-1}; \mathcal{T}(f); \sigma$$

Par abus de notation, notons $\phi : T^\omega \rightarrow \rho X.T$ l'isomorphisme dans **Par** généré par l'isomorphisme de parties ϕ . On a alors :

$$\begin{aligned} \phi; f &= \phi; \chi_{\mathcal{T}}^{-1}; \mathcal{T}(f); \sigma \\ &= \mathcal{T}(\phi; f); \sigma \end{aligned}$$

On utilise l'égalité $\phi; \chi_{\mathcal{T}}^{-1} = \mathcal{T}(\phi)$, qui est triviale par définition de ces stratégies. Après n substitutions de cette expression dans elle-même, on obtient :

$$\phi; f = \mathcal{T}^{n+1}(\phi; f); \mathcal{T}^n(\sigma); \dots; \mathcal{T}(\sigma); \sigma$$

Autrement dit, pour tout $n \in \mathbb{N}$, on a :

$$\mathcal{T}^{n+1}(\phi; f); \sigma^{(n)} = \phi; f$$

Maintenant prenons $s \in \phi; f$. Soit n la profondeur de $s|_{T^\omega}$, c'est-à-dire le n maximal tel que $s|_{T^\omega} \in \mathcal{L}_{T^n(1)}$. Un tel n existe bien, puisque s est finie. On a $s \in \mathcal{T}^{n+1}(\phi; f); \sigma^{(n)}$, mais s est de profondeur n donc $s \in \sigma^{(n)}$, donc $s \in \sigma^{(\omega)}$. Or $\phi; f$ et $\sigma^{(\omega)}$ sont deux stratégies totales, donc par le Lemme 2.1.1, il suffit d'une inclusion pour avoir l'égalité. Par conséquent $f = \phi^{-1}; \sigma^{(\omega)} = \sigma^\ddagger$. \square

Théorème 8.2.11. *Soit \mathcal{T} un foncteur ouvert à gain d'arène ouverte sous-jacente stricte, alors $(\nu X.\mathcal{T}, \chi_{\mathcal{T}}^{-1})$ est une coalgèbre terminale de \mathcal{T} .*

Démonstration. Soit (\mathcal{A}, σ) une autre coalgèbre terminale de \mathcal{T} . Il faut vérifier qu'il existe une unique stratégie gagnante $\sigma^\ddagger : \mathcal{B} \Rightarrow \nu X.\mathcal{T}$ telle que le diagramme suivant commute.

$$\begin{array}{ccc} \mathcal{T}(\nu X.\mathcal{T}) & \xleftarrow{\mathcal{T}(\sigma^\ddagger)} & \mathcal{T}(\mathcal{B}) \\ i_{\mathcal{T}} \uparrow & & \uparrow \sigma \\ \nu X.\mathcal{T} & \xleftarrow{\sigma^\ddagger} & \mathcal{B} \end{array}$$

Comme pour le cas précédent, on donne d'abord une définition directe de $\sigma^{(\omega)}$:

$$\begin{aligned} \sigma^{(1)} &= \sigma \\ \sigma^{(n+1)} &= \sigma^{(n)}; \mathcal{T}^n(\sigma) \\ \sigma^{(\omega)} &= \bigcup_{i \in \mathbb{N}} \sigma|_{\mathcal{B} \Rightarrow T^i} \end{aligned}$$

Puis, si on étend l'isomorphisme de parties $\phi : \mathcal{L}_{\rho X.T}^\omega \rightarrow \mathcal{L}_{T^\omega}^\omega$ à $\mathcal{B} \Rightarrow \rho X.T$ par l'identité sur les coups dans \mathcal{B} , on définit :

$$\sigma^\ddagger = \phi^{-1}(\sigma^{(\omega)})$$

Pour les mêmes raisons que dans le cas des algèbres initiales, on obtient bien une stratégie innocente et totale. Détaillons la preuve qu'elle est gagnante. Comme pour le cas précédent, on surchargera la notation ϕ et ϕ désignera également la stratégie sur $\rho X.T \Rightarrow T^\omega$ donnée par l'isomorphisme de partie ϕ . Soit s éventuellement infinie dans σ^\ddagger , telle que $s|_{\mathcal{B}}$ est gagnante sur \mathcal{B} . Considérons alors $s' = s|_{\nu X.\mathcal{T}}$. Soit elle a une profondeur infinie et s est immédiatement gagnante, soit elle a une profondeur finie. Dans ce cas, avec n sa profondeur, on a $\phi(s) \in \sigma^{(n)}$, or $\sigma^{(n)}$ est une composition de stratégies gagnantes donc est gagnante, donc $\phi(s)$ est une partie gagnante sur $\mathcal{B} \Rightarrow \mathcal{T}^n(1)$, donc c'est également le cas de s sur $\mathcal{B} \Rightarrow \nu X.\mathcal{T}$ par définition du gain sur $\nu X.\mathcal{T}$.

La preuve que σ^\ddagger est bien la seule stratégie gagnante satisfaisant cette contrainte est exactement duale de celle pour le cas de l'algèbre initiale, elle ne sera donc pas répétée. \square

8.2.4 Une catégorie μ -fermée

La structure introduite jusqu'à maintenant permet de donner dans **Par** des invariants minimaux aux foncteurs ouverts, ainsi que des algèbres initiales / coalgèbres finales dans **Gain** aux foncteurs ouverts à gain. Cette structure permet d'interpréter, par exemple, les types inductifs ou coinductifs *non entrelacés*, c'est-à-dire tels qu'il n'y a qu'une seule variable libre à la fois lors de la construction des formules correspondantes. Dans cette sous-section,

on va apporter la dernière pierre à la structure μ -fermée de **Gain** en montrant comment on peut donner une structure de foncteurs ouverts à gain aux *algèbres initiales paramétrés* et *coalgèbres terminales paramétrées*, permettant ainsi d'interpréter n'importe quelle formule de μLJ , même entrelacée.

Proposition 8.2.12. *Soit $\mathcal{T} : \mathbf{Gain} \times \mathfrak{D} \rightarrow \mathbf{Gain}$ un foncteur ouvert à gain, où \mathfrak{D} est un produit de copies de **Gain** et **Gain**^{op}. Alors l'algèbre initiale paramétrée :*

$$\mathcal{T}^\mu : \mathfrak{D} \rightarrow \mathbf{Gain}$$

est un foncteur ouvert à gain.

Démonstration. Rappelons que \mathcal{T}^μ est défini comme l'unique foncteur dans $\mathfrak{D} \rightarrow \mathbf{Gain}$ tel que pour tout D objet de \mathfrak{D} on ait $\mathcal{T}^\mu(D) = \mu X.T(X, D)$, et tel que $\chi_D : \mathcal{T}(\mathcal{T}^\mu(D), D) \rightarrow \mathcal{T}^\mu(D)$ soit naturel en D . Plus précisément, si $f : D_1 \rightarrow D_2$, $\mathcal{T}^\mu(f)$ est défini comme l'unique morphisme de $\mathcal{T}(-, D_1)$ -algèbre de l'algèbre initiale $(\mathcal{T}^\mu(D_1), \chi_{D_1})$ vers l'algèbre $(\mathcal{T}^\mu(D_2), \mathcal{T}(\mathcal{T}^\mu(D_2), f); \chi_{D_2})$. Dans la syntaxe utilisée dans la preuve d'initialité de $(\mu X.T, \chi_T)$, cela signifie :

$$\mathcal{T}^\mu(f) = (\mathcal{T}(\mathcal{T}^\mu(D_2), f); \chi_{D_2})^\dagger$$

Tout d'abord, il faut montrer que l'action de $\mathcal{T}^\mu(f)$ sur les stratégies est donnée par un foncteur ouvert, dont l'arène ouverte génératrice sera $\rho X.T$. Montrons d'abord que toute P -vue est P -fonctorielle, afin de pouvoir procéder aux restrictions verticales. Soit donc s une P -vue de $\mathcal{T}^\mu(f)$, donc de $(\mathcal{T}(\mathcal{T}^\mu(D_2), f); \chi_{D_2})^\dagger$, soit $(\mathcal{T}(\mathcal{T}^\mu(D_2), f); \chi_{D_2})^{(\omega)}$ à isomorphisme de parties près (on va garder silencieux l'usage de cet isomorphisme pour clarifier la preuve). Donc, il existe $n \in \mathbb{N}$ tel que s est une P -vue de $(\mathcal{T}(\mathcal{T}^\mu(D_2), f); \chi_{D_2})^{(n)}$. Montrons donc que s est P -fonctorielle par induction sur n . Si $n = 0$, on a $s \in \mathcal{T}(\mathcal{T}^\mu(D_2), f); \chi_{D_2}$, donc est bien P -fonctorielle pour chaque composante de \mathfrak{D} puisqu'obtenu par l'action d'un foncteur ouvert, donc une stratégie fonctorielle. Sinon, si $\alpha = \mathcal{T}(\mathcal{T}^\mu(D_2), f); \chi_{D_2}$, on a $s \in \alpha^{(n+1)} = \mathcal{T}^n(\alpha, D_1); \alpha^{(n)}$ donc est une P -vue d'une composition de stratégies fonctorielles en toutes les composantes de \mathfrak{D} , donc P -fonctorielle. Par le même raisonnement, on montre que s restreint à sa composante fonctorielle est dans $id_{\mu X.T}$, et que sa restriction aux différentes composantes argument de $D_1 \Rightarrow D_2$ sont dans les différentes composantes de f . C'est trivial si $s \in \alpha$ puisque α est donnée par composition verticale. Si $s \in \mathcal{T}^n(\alpha, D_1); \alpha^{(n)}$, alors c'est une conséquence de l'hypothèse d'induction et de la loi d'échange pour la composition verticale.

Finalement, montrons que les forces du foncteur ouvert $\mu X.T$ sont gagnantes. Pour cela, décomposons $\mathfrak{D} = \mathfrak{C} \times \mathfrak{D}'$, où $\mathfrak{C} = \mathbf{Gain}$ ou $\mathfrak{C} = \mathbf{Gain}^{op}$, et soit D un objet de \mathfrak{D}' , il faut montrer que les stratégies suivantes sont gagnantes, en fonction de la variance de \mathfrak{C} .

$$\begin{aligned} \theta_{\Gamma, \mathcal{A}}^{(\mu X.T)(-, D)} & : \Gamma \times ((\mu X.T)(\mathcal{A}, D) \rightarrow ((\mu X.T)(\Gamma \times \mathcal{A}, D) \\ \rho_{\Gamma, \mathcal{A}}^{(\mu X.T)(-, D)} & : \Gamma \times ((\mu X.T)(\Gamma \times \mathcal{A}, D) \rightarrow ((\mu X.T)(\mathcal{A}, D) \end{aligned}$$

Soit $s \in \theta_{\Gamma, \mathcal{A}}^{(\mu X.T)(-, D)}$, supposons que s est gagnante sur $\Gamma \times ((\mu X.T)(\mathcal{A}, D)$. Cela signifie que $s|_\Gamma$ est gagnante, ainsi que $s|_{((\mu X.T)(\mathcal{A}, D)}$. Cette deuxième hypothèse, étant donné la définition du gain sur $(\mu X.T)(\mathcal{A}, D)$, implique que s est de profondeur n . Mais dans ce cas on constate immédiatement que $s \in \theta^{\mathcal{T}^n(\mathbf{1}, -, D)}$ (à isomorphisme de parties près), et est donc gagnante puisque $\mathcal{T}^n(\mathbf{1}, -, D)$ est un foncteur ouvert à gain. De même, soit $s \in \rho_{\Gamma, \mathcal{A}}^{(\mu X.T)(-, D)}$. La preuve est similaire. Puisqu'on suppose s gagnante à gauche, celle-ci doit être à profondeur finie n , et est donc une partie de la force d'un foncteur ouvert à gain, donc gagnante. \square

Proposition 8.2.13. *Soit $\mathcal{T} : \mathbf{Gain} \times \mathfrak{D} \rightarrow \mathbf{Gain}$ un foncteur ouvert à gain, où \mathfrak{D} est un produit de copies de \mathbf{Gain} et \mathbf{Gain}^{op} . Alors la coalgèbre terminale paramétrée :*

$$\mathcal{T}^\nu : \mathfrak{D} \rightarrow \mathbf{Gain}$$

est un foncteur ouvert à gain.

Démonstration. La preuve est fortement similaire. En effet, dans notre syntaxe, cette coalgèbre terminale paramétrée est définie par, si $f : D_1 \rightarrow D_2$:

$$\mathcal{T}^\nu(f) = (\chi_{D_1}^{-1}; \mathcal{T}(\mathcal{T}^\nu(D_1), f))^\ddagger$$

En passant sous silence l'usage de l'isomorphisme de parties, toutes les parties de $\mathcal{T}^\nu(f)$ sont donc des parties de $(\chi_{D_1}^{-1}; \mathcal{T}(\mathcal{T}^\nu(D_1), f))^{(\omega)}$ donc (pour $n \in \mathbb{N}$) de $(\chi_{D_1}^{-1}; \mathcal{T}(\mathcal{T}^\nu(D_1), f))^{(n)}$. On montre donc par récurrence sur n que si s est une telle P -vue, s est bien P -fonctorielle en toutes les composantes de \mathfrak{D} . On en déduit également que c'est bien une partie de $(\rho X.T)(f)$.

Il faut également montrer que si on décompose $\mathfrak{D} = \mathfrak{C} \times \mathfrak{D}'$, où \mathfrak{C} est \mathbf{Gain} ou \mathbf{Gain}^{op} , et si D est un objet de \mathfrak{D}' , que les forces (selon la variance de \mathfrak{C} :

$$\begin{aligned} \theta_{\Gamma, \mathcal{A}}^{(\nu X.T)(-, D)} & : \Gamma \times ((\nu X.T)(\mathcal{A}, D) \rightarrow ((\nu X.T)(\Gamma \times \mathcal{A}, D) \\ \rho_{\Gamma, \mathcal{A}}^{(\nu X.T)(-, D)} & : \Gamma \times ((\nu X.T)(\Gamma \times \mathcal{A}, D) \rightarrow ((\nu X.T)(\mathcal{A}, D) \end{aligned}$$

sont gagnantes. Mais si on prend s éventuellement infinie dans $\theta_{\Gamma, \mathcal{A}}^{(\nu X.T)(-, D)}$, alors soit s est gagnante à droite et il n'y a rien à montrer, soit elle est perdante à droite. Mais dans ce cas, par la définition du gain sur $((\nu X.T)(\Gamma \times \mathcal{A}, D)$, c'est qu'elle a nécessairement une profondeur finie n . Dans ce cas, c'est à isomorphisme de parties près une partie de $\theta^{\mathcal{T}^n(\mathbf{1}, -, D)}$, auquel cas elle est bien gagnante puisque \mathcal{T} est un foncteur ouvert à gain. La preuve est identique pour ρ . \square

Puisqu'on sait maintenant que les algèbres initiales paramétrées et les coalgèbres terminales paramétrées sont bien des foncteurs ouverts à gain, on peut énoncer le théorème principal de cette partie :

Théorème 8.2.14. *\mathbf{Gain} est μ -fermée pour les types stricts et pseudo-polynomiaux, et donc un modèle du fragment strict et pseudo-polynomial de μLJ .*

Démonstration. Trivial, par les développements précédents et par le Théorème 6.4.2. \square

En particulier, on en déduit le corollaire suivant :

Corollaire 8.2.15. *Le fragment strict et pseudo-polynomial de μLJ est cohérent, c'est-à-dire qu'il n'y a pas de preuve de 0.*

Démonstration. La formule 0 est interprétée par le jeu à gain $\mathbf{0}$ dont l'arène sous-jacente n'a qu'un seul coup \circ , de polarité Opposant. Joueur n'a pas de réponse possible à ce coup, par conséquent il ne peut y avoir de stratégie totale dans le jeu $\mathbf{0}$. Or on a montré que toute preuve de μLJ de la formule 0 s'interpréterait dans une stratégie totale (et même gagnante) sur le jeu $\mathbf{0}$, donc il n'y a pas de telle preuve. \square

Évidemment, le Théorème 8.2.14 est plus important que son corollaire : d'abord parce qu'on connaît les limites que le second théorème d'incomplétude de Gödel pose à de tels énoncés, mais aussi parce que μLJ se veut une logique conçue pour interpréter le calcul : en ce sens, un théorème de normalisation ou d'élimination des coupures serait plus intéressant qu'un théorème de cohérence. C'est ce point qui va être discuté dans cette dernière section.

8.3 Sur la complétude

La sémantique de jeux s'est distinguée depuis deux décennies par sa propension à fournir des modèles *pleinement complets* [AJ94] (où chaque morphisme est l'interprétation d'une preuve) voire pleinement abstraits (chaque morphisme est l'interprétation d'une unique preuve à équivalence observationnelle près). Il est donc naturel, après la construction de ce modèle, de s'interroger sur sa complétude : a-t-on trouvé une façon alternative mais équivalente de voir les preuves de μLJ , ou bien a-t-on récupéré au passage des «preuves» supplémentaires ? Malheureusement, la deuxième réponse est la bonne : si $\Gamma \vdash A$ est un séquent, il y a potentiellement beaucoup plus de stratégies gagnantes et totales dans $\llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket$ que de preuves de $\Gamma \vdash A$. En effet, les stratégies totales gagnantes peuvent être des objets infinis sans contrainte de récursivité :

- Certaines stratégies totales gagnantes peuvent calculer des fonctions non récursives, par exemple la stratégie $\kappa : \mathbf{nat} \Rightarrow \mathbf{bool}$ qui décide le problème de l'arrêt pour la n -ième machine de Turing (où n est l'entier donné en argument) est une stratégie gagnante acceptée dans le modèle.
- Certaines stratégies peuvent calculer des fonctions récursives, mais être non récursives en tant qu'ensemble de parties.

Pourtant, rien n'empêche dans **Gain** d'appliquer le processus usuel de reconstruction de termes à partir de stratégies innocentes (Section 2.3.2). Pour chaque stratégie σ , on récupèrera ainsi une «preuve» π antécédente de σ pour la fonction d'interprétation, mais on ne pourra pas assurer la finitude de π . Une conséquence intéressante en sera qu'on peut éliminer les coupures dans toute preuve de μLJ , au prix d'un développement infini.

Preuves infinies. Les formules de la logique μLJ^ω sont les mêmes que celles de μLJ , et les «preuves» sont les arbres éventuellement infinis construits à partir du groupe identité (sans la coupure), du groupe logique, et des règles μ_r et ν_l . Les besoins de cette sous-section demandent également l'ajout de la règle suivante :

$$\frac{\Gamma \vdash T(\nu X.T(X))}{\Gamma \vdash \nu X.T(X)} \text{cov}_r$$

Elle est définissable dans μLJ comme garanti par la Proposition 1.3.1, mais cette définition utilise une coupure, alors qu'elle s'interprète comme μ_r , c'est-à-dire comme l'identité (à isomorphisme de parties près) dans la sémantique de jeux.

Proposition 8.3.1. *Si π est une «preuve» infinie d'un séquent $\Gamma \vdash A$, alors $\llbracket \pi \rrbracket$ est une stratégie innocente, totale et bien parenthésée sur $\llbracket \Gamma \rrbracket \Rightarrow \llbracket A \rrbracket$.*

Démonstration. On peut voir π comme une borne supérieure d'une suite croissante de preuves partielles préfixes de π . Notons cette séquence π_1, π_2, \dots . Pour chaque π_i , $\llbracket \pi_i \rrbracket$ est une stratégie innocente et bien parenthésée telle que $\llbracket \pi_i \rrbracket \subseteq \llbracket \pi_{i+1} \rrbracket$. Cette suite $\llbracket \pi_i \rrbracket$ est non stagnante, c'est-à-dire qu'elle croît strictement infiniment souvent. En effet, les règles de μLJ (à part bien sûr la coupure) qui ne font pas croître strictement l'interprétation sont \times_l et \Rightarrow_r , qu'on ne peut pas rencontrer infiniment souvent puisqu'elles décomposent des connecteurs et que les séquents sont finis. Ainsi cette séquence $\llbracket \pi_i \rrbracket$ admet une borne supérieure $\llbracket \pi \rrbracket$, qui est infinie si π était infinie et qui coïncide avec l'interprétation habituelle sinon. \square

Remarque. Bien sûr, cette procédure ne définit pas nécessairement des stratégies gagnantes, il suffit pour s'en rendre compte de considérer l'interprétation de la «preuve» infinie définie par l'équation réursive suivante :

$$\pi = \frac{\frac{\pi}{\vdash \mathbf{nat}} \xrightarrow{+r}}{\vdash 1 + \mathbf{nat}} \xrightarrow{\mu_r} \frac{\pi}{\vdash \mathbf{nat}}$$

Il s'agit de l'«entier» infini, qui est toujours le successeur de lui-même. D'après la proposition précédente, $\llbracket \pi \rrbracket$ donne une stratégie innocente, bien parenthésée et totale sur l'arène $\llbracket \mathbf{nat} \rrbracket$, mais qui ne peut clairement pas être gagnante.

Définissabilité. Soit maintenant $\Gamma \vdash A$ un séquent, et $\sigma : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket$ une stratégie gagnante. Il est immédiat d'adapter la procédure de définissabilité aux formules de μLJ : à chaque fois qu'on doit définir un terme de type $\mu X.T$ ou $\nu X.T$, on applique la règle μ_r ou $co - \nu_r$ et on applique la définissabilité sur $T(\mu X.T)$ ou $T(\nu X.T)$, dont le connecteur principal est maintenant un connecteur logique. La validité de cette procédure revient à celle de la procédure de définissabilité habituelle. Bien sûr, elle ne termine pas mais converge vers une «preuve» infinie σ^\bullet dans μLJ^ω du séquent $\Gamma \vdash A$. Notons que cette «preuve» a d'un point de vue logique autant de valeur qu'une preuve usuelle de μLJ , puisque par construction son interprétation dans **Gain** donne une stratégie gagnante.

Conséquences. Énonçons les deux résultats suivants, conséquences de cette procédure de définissabilité. Le premier énonce une sorte d'élimination des coupures de μLJ , au prix du passage dans la syntaxe infinie. Notons que cette élimination des coupures ne se fera que vis à vis de l'équivalence générée par le modèle :

$$\pi \simeq_{\mathbf{Par}} \pi' \Leftrightarrow \llbracket \pi \rrbracket = \llbracket \pi' \rrbracket$$

Et non pas vis-à-vis de l'équivalence générée par la réduction sur les preuves, qui bien sûr ne pourrait en aucun cas mettre en relation une preuve finie avec une preuve infinie.

Théorème 8.3.2. *Soit π d'un séquent $\Gamma \vdash A$ de μLJ , alors il existe une «preuve» π' éventuellement infinie, du même séquent $\Gamma \vdash A$ dans μLJ^ω , telle que $\pi \simeq_{\mathbf{Par}} \pi'$.*

Démonstration. On définit $\pi' = \llbracket \pi \rrbracket^\bullet$. □

Bien sûr, si les formules du séquent ne contiennent pas de points fixes, il est possible d'éliminer complètement les coupures tout en restant dans μLJ :

Proposition 8.3.3. *Soit π une preuve dans μLJ d'un séquent $\Gamma \vdash A$, tel que ni A ni une formule de Γ ne contiennent d'occurrences de μ ou ν (même s'il y en a des occurrences dans π , elles n'apparaissent pas dans le séquent final). Alors il existe une preuve π' de μLJ sans coupures telle que $\pi \simeq_{\mathbf{Par}} \pi'$.*

Démonstration. Par induction, il est immédiat que $\llbracket \Gamma \rrbracket \Rightarrow \llbracket A \rrbracket$ est un jeu à gain d'arène bien fondée, et respectant le principe de justice (puisque les unités le respectent). Par la Proposition 8.1.15, les stratégies gagnantes sont donc noéthériennes. Or par induction immédiate sur Γ et A on vérifie que l'arène ambiante est finie, donc par le Théorème 4.3.6 les stratégies gagnantes sont finies, donc la procédure de définissabilité termine sur une preuve π' de μLJ . Par construction de π' , on a bien $\pi' \simeq_{\mathbf{Par}} \pi$. □

Notons que la proposition précédente s'étend directement aux séquents $\Gamma \vdash A$ tels que μ n'apparaît pas positivement dans Γ et que ν n'apparaît pas positivement dans A . En effet, le gain permet alors de convertir les occurrences présentes de μ et ν en des dépliages finis de foncteurs, ainsi on se retrouve dans le cas précédent. Notons également que c'est un véritable résultat de normalisation par évaluation effectif des preuves de μLJ sans point fixe dans le type externe. Par ailleurs, les intuitions de la Partie I laissent croire qu'il correspond au calcul de la forme normale obtenue pour une certaine notion de réduction linéaire de tête.

Fidélité. Lorsqu'on a un modèle de jeux, il est naturel de se poser la question de sa *fidélité*, c'est-à-dire de savoir si le foncteur d'interprétation est fidèle. Pour ce système, ce n'est malheureusement pas le cas. La syntaxe de μLJ étant peu intuitive, donnons le contre-exemple dans un langage de programmation à la CamL. Considérons les deux termes suivants :

```
let rec iter f n b =
  if n = 0 then b
  else iter f (n-1) (f b)
```

```
let rec iter' f n b =
  if n = 0 then b
  else not (iter' f (n-1) (f (not b)))
```

Il est immédiat de voir (par induction sur n) que sur toutes entrées f , n et b , `not (iter f n (not b))` et `iter' f n b` renvoient le même résultat. En fait, ces deux fonctions (vues comme des termes de μLJ) ont le même dépliage infini dans μLJ^ω : le dépliage de `iter'` fait apparaître des séquences `not (not (-))` qui se simplifient immédiatement, `not` étant son propre inverse. Or ce dépliage infini caractérise la stratégie correspondante, donc ces deux termes ont la même interprétation dans **Gain**. Cependant, ils ne sont certainement pas réductibles l'un à l'autre : tout ce qui est dans un `let rec` serait en μLJ derrière une règle μ_l , et toutes les réductions préservent les blocs μ_l . Un problème intéressant serait alors de savoir s'il existe une réduction sur μLJ dont la théorie coïncide avec l'égalité dans le modèle de jeux, malheureusement il nous faudra remettre cela à plus tard.

Conclusions et perspectives

Pour conclure, mentionnons pour chacune des parties de cette thèse les questions ouvertes et axes de recherches soulevés.

Première partie

Masquage et β -réduction. La notion d'agent introduite en Section 4.2.2 donne sur les structures de pointeurs une notion adéquate de *stratégie non β -normale*. Bien sûr, il s'agit d'une situation extrêmement restreinte où on n'a que très peu d'information sur le comportement des stratégies. Cependant, il semble que la définition d'agent puisse se généraliser naturellement à un cadre plus habituel, où les stratégies sont innocentes et données par un ensemble de P -vues. Ainsi qu'on l'a mentionné à la fin de la Section 4.2.3, il semblerait que la réduction des coups orphelins utilisée pour la troisième preuve du théorème de finitude ait alors, sur ces agents généralisés, un effet ressemblant à une forme restreinte de β -réduction. C'est intéressant car il semble par ailleurs que le mécanisme de masquage usuel en sémantique de jeux puisse se décomposer en étapes élémentaires de cette opération d'élimination des coups orphelins. On a donc un espoir de décomposer la composition des stratégies innocentes en une étape de composition parallèle et une séquence d'élimination des coups orphelins à voir comme une séquence de β -réductions.

Longueur de la réduction linéaire de tête. À la connaissance de l'auteur, le problème de la longueur des chaînes de réduction linéaire de tête (pour des termes simplement typés) n'a jamais été traité de manière satisfaisante. C'est pourtant une question intéressante, de par les relations entre la réduction linéaire de tête et l'exécution des machines abstraites. On a présenté en Section 4.2.2 une réduction élémentaires sur les arbres d'entiers en relation très forte avec les structures de pointeurs, donc également avec les chaînes de réduction linéaire de tête de termes du λ -calcul simplement typé. Cette approche purement combinatoire, débarrassée de tous les aspects liés à la syntaxe et au typage, s'annonce prometteuse pour attaquer ce problème.

Modélisation des types dépendants. Initialement, les théorèmes de finitude développées dans la première partie avaient pour objectif d'assurer la construction d'une catégorie de stratégies totale sur laquelle baser un modèle des types dépendants. Ces travaux sont en bonne voie, même s'ils n'ont pas pu être présentés ici faute de temps.

Deuxième partie

Conjecture de renforcement. Comme on l'a vu au Chapitre 6, la représentation des règles étendues (*i.e.* avec contexte pour les foncteurs et les points fixes) de μLJ impose la construction d'une structure de forces complexe, assurant aussi bien le relèvement des foncteurs syntaxiques que l'uniformité de ce relèvement. On aimerait pouvoir se passer de cette structure et le meilleur moyen pour cela serait de montrer la Conjecture 6.2.10. On peut imaginer plusieurs pistes, mais la complexité des égalités à montrer et la répétitivité des preuves requises laissent supposer qu'elles pourraient être en partie automatisées, par exemple dans un assistant de preuves comme COQ.

Foncteurs ouverts et polymorphisme. On a introduit au Chapitre 7 la classe des foncteurs ouverts, afin de modéliser les types à variables libres. Cette classe de foncteurs est riche en propriétés. D'une part, les foncteurs ouverts viennent avec des notions correspondantes de transformations naturelles ouvertes et de transformations dinaturelles ouvertes, ainsi qu'avec des lois d'échange les reliant aux foncteurs ouverts. D'autre part, on a montré (Théorème 7.3.18) que toutes les transformations naturelles entre foncteurs ouverts étaient en fait des transformations naturelles ouvertes. Ce résultat puissant permettrait par exemple de montrer un résultat de définissabilité sur les transformations naturelles entre arènes ouvertes finies. Tout ceci semble tracer les contours d'une modélisation très élégante du système F à la Curry, qui semble différente des approches précédentes de Abramsky, Hughes et De Lataillade dont la cible principale était la variante à la Church. Pour cela, il faudra cependant commencer par étendre la définition des foncteurs ouverts à une classe plus large de foncteurs ouverts et redémontrer le Théorème 7.3.18 pour les transformations dinaturelles ouvertes.

Liens avec les jeux sur les graphes. On se trouve ici dans une situation où, modulo les difficultés sémantique pour donner sens à la notion de foncteur ouvert, les conditions de gain utilisées dans les jeux de parité s'importent tout naturellement dans le cadre de la sémantique de jeux. On peut se demander dans quelle mesure cette situation est générale, par exemple dans quelle mesure des conditions de gains quelconques sur les parties infinies dans un graphe génèrent librement des conditions de gain sur les parties à pointeurs. On peut ensuite se demander dans quelle mesure de tels points pourraient permettre l'import de résultats dans les jeux utilisés en vérification jusqu'en sémantique de jeux et réciproquement.

Complétude. On pourrait finalement chercher à rapprocher encore le modèle de jeux de la syntaxe. Cela peut se faire de plusieurs façons. D'abord, il serait intéressant de rechercher une caractérisation sémantique des stratégies gagnantes et totales obtenues par l'interprétation de preuves de μLJ : elles sont au moins récursives, mais il nous faut trouver d'autres conditions. Par ailleurs, on a vu que l'égalité dans le modèle de jeux ne coïncide pas avec la relation de convertibilité dans la syntaxe. De notre point de vue, le défaut ici n'est pas dans notre sémantique, mais plutôt dans la syntaxe. Ainsi on aimerait donner un algorithme décidant en temps fini, si π et π' sont deux preuves de μLJ , si leurs interprétations sont égales. Si on y parvient, on aimerait de plus savoir si cette équivalence est internalisable par l'ajout de réductions supplémentaires dans la syntaxe, par exemple sous la forme de commutations.

Isomorphismes. Dans [Lau05], Laurent a utilisé la sémantique de jeux (notamment les résultats présentés en Section 2.4.2) pour retrouver les équations engendrant les isomorphismes de types en logique classique d'abord découverts par Di Cosmo [Cos05]. Cette approche a été ensuite étendue au second ordre par De Lataillade [Lat08]. À la connaissance de l'auteur, le problème des isomorphismes de types n'a pas été traité pour les types inductifs et coinductifs. Il est donc naturel d'espérer appliquer les outils développés ici à leur étude. Une question attenante est celle des différentes hiérarchies de types de μLJ , comme étudiées par exemple par Belkhir et Santocanale [BS08] pour ce qu'on a appelé ici le μ -calcul propositionnel. De façon élémentaire, le problème est de savoir si toute formule de μLJ peut être définie à isomorphisme près par une formule où le nombre d'alternances entre μ et ν (hiérarchie d'alternance) ou le nombre de variables liées par μ ou ν (hiérarchie de variables) est borné. L'extrême précision de la caractérisation des isomorphismes en sémantique de jeux donnée par le théorème de Laurent laisse croire que la présence de l'implication ne change rien aux résultats connus pour le μ -calcul propositionnel, mais on laisse cela pour des travaux ultérieurs.

Gain sur les P -vues. Si s est une partie infinie à profondeur finie, on a montré dans la Partie I qu'elle admettait nécessairement une fourche consciente infinie. Autrement dit, on peut en extraire une P -vue ou une O -vue perdante pour le principe de justice. Il est naturel d'examiner des généralisations de ce résultat aux parties à profondeur infinie : si s est une partie infinie perdante pour un jeu à gain \mathcal{A} , peut-on extraire de s une P -vue infinie perdante ? Autrement dit, suffit-il d'examiner les P -vues d'une stratégie $\sigma : \mathcal{A}$ afin de déterminer si elle est gagnante ? Outre le caractère fort naturel de cette question, elle s'avère cruciale pour généraliser le présent modèle de jeux à un calcul des séquents tel que ID1 [BFPS81], où les preuves infinies sont autorisées mais contrôlées par l'intermédiaire d'une notion de gain sur les branches infinies, fort similaire à la nôtre.

Bibliographie

- [AA99] Andreas Abel and Thorsten Altenkirch. A predicative strong normalisation proof for a lambda-calculus with interleaving inductive types. In *TYPES*, pages 21–40, 1999.
- [Abr03] Samson Abramsky. Sequentiality vs. concurrency in games and logic. *Mathematical Structures in Computer Science*, 13(4) :531–565, 2003.
- [AHM98] S. Abramsky, K. Honda, and G. McCusker. A fully abstract game semantics for general references. In *13th IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1998.
- [AJ94] Samson Abramsky and Radha Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59(2) :543–574, 1994.
- [AJM00] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Information and Computation*, 163(2) :409–470, 2000.
- [AM97] S. Abramsky and G. McCusker. Linearity, Sharing and State : a Fully Abstract Game Semantics for Idealized Algol with active expressions, 1997.
- [AM98] Samson Abramsky and Guy McCusker. Call-by-value games. In Mogens Nielsen and Wolfgang Thomas, editors, *6th Annual Conference of the European Association for Computer Science Logic*, volume 1414 of *Lecture Notes in Computer Science*. Springer, 1998.
- [AN01] A. Arnold and D. Niwiński. *Rudiments of μ -calculus*. North Holland, 2001.
- [Bar85] Henk Barendregt. *The Lambda Calculus : Its Syntax and Semantics*. North Holland, 1985.
- [Bec69] J. Beck. Distributive laws. In *Seminar on triples and categorical homology theory*, volume 80, pages 119–130, 1969.
- [BFPS81] W. Buchholz, S. Feferman, W. Pohlers, and W. Sieg. Iterated inductive definitions and subsystems of analysis : recent proof-theoretical studies. 1981.
- [BFSS90] E. S. Bainbridge, Peter J. Freyd, Andre Scedrov, and Philip J. Scott. Functorial polymorphism. *Theoretical Computer Science*, 70(1) :35–64, 1990.
- [Bla72] Andrea Blass. Degrees of indeterminacy of games. *Foundations of Mathematics*, 77 :151–166, 1972.
- [Bla92] Andrea Blass. A games semantics for linear logic. *Annals of Pure and Applied Logic*, 56 :183–220, 1992.
- [Blu08] William Blum. *The Safe λ -calculus*. PhD thesis, Oxford University, 2008.
- [BM07] David Baelde and Dale Miller. Least and greatest fixed points in linear logic. In *LPAR*, pages 92–106, 2007.

- [BS08] Walid Belkhir and Luigi Santocanale. The variable hierarchy for the lattice $\mathbb{N}^{\mathbb{N}}$. In *LPAR*, pages 605–620, 2008.
- [CD94] Thierry Coquand and Peter Dybjer. Inductive definitions and type theory : an introduction (preliminary version). In *FSTTCS*, pages 60–76, 1994.
- [CF92] R. Cockett and T. Fukushima. About charity. Technical report, 1992.
- [CH98] Pierre-Louis Curien and Hugo Herbelin. Computing with Abstract Böhm Trees. In *Fuji International Symposium on Functional and Logic Programming*, pages 20–39, 1998.
- [CH00] Pierre-Louis Curien and Hugo Herbelin. The duality of computation. In *International Conference on Functional Programming (ICFP)*, pages 233–243, 2000.
- [CH09] Pierre Clairambault and Russ Harmer. Totality in arena games. *Annals of Pure and Applied Logic*, 2009.
- [CHP04] E. Cheng, M. Hyland, and J. Power. Pseudo-distributive laws. *Electronic Notes in Theoretical Computer Science*, 83, 2004.
- [Cla09a] Pierre Clairambault. Least and greatest fixpoints in game semantics. In *FOSSACS*, pages 16–31, 2009.
- [Cla09b] Pierre Clairambault. Least and greatest fixpoints in game semantics. In *FICS*, 2009.
- [Coq95] Thierry Coquand. A semantics of evidence for classical arithmetic. *Journal of Symbolic Logic*, 60(1) :325–337, 1995.
- [Cos05] Roberto Di Cosmo. A short survey of isomorphisms of types. *Mathematical Structures in Computer Science*, 15(5) :825–838, 2005.
- [CS92] J.R.B. Cockett and D. Spencer. Strong categorical datatypes I. In *Category theory 1991 : proceedings of an International Summer Category Theory Meeting, held June 23-30, 1991*, page 141. Canadian Mathematical Society, 1992.
- [CS95] J. Robin B. Cockett and Dwight Spencer. Strong Categorical Datatypes II : A Term Logic for Categorical Programming. *Theoretical Computer Science*, 139(1&2) :69–113, 1995.
- [Cur98] Pierre-Louis Curien. Abstract Böhm trees. *Mathematical Structures in Computer Science*, 8(6) :559–591, 1998.
- [DB87] NG De Bruijn. Generalizing Automath by means of a lambda-typed lambda calculus. *Mathematical Logic and Theoretical Computer Science*, 106 :71–92, 1987.
- [DHR96a] Vincent Danos, Hugo Herbelin, and Laurent Regnier. Game semantics & abstract machines. In *11th IEEE Symposium on Logic in Computer Science*, pages 394–405, 1996.
- [DHR96b] Vincent Danos, Hugo Herbelin, and Laurent Regnier. Games semantics and abstract machines. In *11th IEEE Symposium on Logic in Computer Science*, pages 394–405, New Brunswick, 1996. IEEE Computer Society Press.
- [DR] Vincent Danos and Laurent Regnier. How abstract machines implement head linear reduction. *Higher Order and Symbolic Computation (to appear)*.
- [DV87] RC De Vrijer. Exactly estimating functionals and strong normalization. *Indagationes Mathematicae*, 49(4) :479–493, 1987.

- [ER03] Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309(1-3) :1–41, 2003.
- [FH02] Claudia Faggian and Martin Hyland. Designs, disputes and strategies. In *Annual Conference of the European Association for Computer Science Logic*, pages 442–457, 2002.
- [Fre90] Peter J. Freyd. Recursive types reduced to inductive types. In A. Carboni, M. C. Pedicchio, and G. Rosolini, editors, *Proceedings of the Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 498–507, Philadelphia, Pennsylvania, June 1990. IEEE Computer Society Press.
- [Fre91] Peter J. Freyd. Algebraically complete categories. In Aurelio Carboni, M. C. Pedicchio, and G. Rosolini, editors, *Proceedings of the 1990 Como Category Theory Conference*, volume 1488 of *Lecture Notes in Mathematics*, pages 131–156. Springer-Verlag, 1991.
- [Fre92] Peter J. Freyd. Remarks on algebraically compact categories. In M.P. Fourman, P.T. Johnson, and A.M. Pitts, editors, *Applications of Categories in Computer Science*, volume 177 of *London Mathematical Society Lecture Note Series*, pages 95–106. Cambridge University Press, 1992.
- [Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50 :1–102, 1987.
- [Gir89] Jean-Yves Girard. Geometry of interaction I : interpretation of system F. In *Logic Colloquium 88*, pages 221–260, Amsterdam, July 1989. North Holland.
- [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, 2001.
- [GTL89] J.Y. Girard, P. Taylor, and Y. Lafont. *Proofs and types*. Citeseer, 1989.
- [Har99] Russ Harmer. *Games and Full Abstraction for Nondeterministic Languages*. PhD thesis, University of London, 1999.
- [Har07] R. Harmer. Innocent game semantics. *Lecture notes*, 2004–2007.
- [HHM07] Russ Harmer, Martin Hyland, and Paul-André Melliès. Categorical combinatorics for innocent strategies. In *IEEE Symposium on Logic in Computer Science*, pages 379–388, 2007.
- [HM99] Russell Harmer and Guy McCusker. A fully abstract game semantics for finite nondeterminism. In *IEEE Symposium on Logic in Computer Science*, pages 422–430, 1999.
- [HO92] Martin Hyland and Luke Ong. Fair games and full completeness for multiplicative linear logic without the mix rule. Manuscrit, 1992.
- [HO00] Martin Hyland and Luke Ong. On full abstraction for PCF : I, II and III. *Information and Computation*, 163(2) :285–408, December 2000.
- [How80] W.A. Howard. The formulae-as-types notion of construction. *To HB Curry : essays on combinatory logic, lambda calculus and formalism*, pages 479–490, 1980.
- [Hug97] Dominic J. D. Hughes. Games and Definability for System F. In *IEEE Symposium on Logic in Computer Science*, pages 76–86, 1997.
- [Hy197] Martin Hyland. Game semantics. In *Semantics and logics of computation*, Publications of the Newton Institute. Cambridge University Press, 1997.

- [Joy77] André Joyal. Remarques sur la théorie des jeux à deux personnes. *Gazette des Sciences Mathématiques du Québec*, 1(4) :46–52, 1977. English version by Robin Houston available.
- [JR97] B. Jacobs and J. Rutten. A tutorial on (co) algebras and (co) induction. *EATCS Bulletin*, 1997.
- [JS88] A. Joyal and R. Street. Braided monoidal categories. *Mathematics Reports*, 86008, 1988.
- [KNO02] Andrew D. Ker, Hanno Nickau, and C.-H. Luke Ong. Innocent game models of untyped lambda-calculus. *Theoretical Computer Science*, 272(1-2) :247–292, 2002.
- [Kri85] J.L. Krivine. Un interpréteur du λ -calcul. *Unpublished*, 1985.
- [Lai97] James Laird. Full abstraction for functional languages with control. In *IEEE Symposium on Logic in Computer Science*, pages 58–67, 1997.
- [Lai01] James Laird. A fully abstract game semantics of local exceptions. In *IEEE Symposium on Logic in Computer Science*, pages 105–114, 2001.
- [Lai05] Jim Laird. A game semantics of the asynchronous π -calculus. In *CONCUR*, pages 51–65, 2005.
- [Lam68] J. Lambek. A fixpoint theorem for complete categories. *Mathematische Zeitschrift*, 103(2) :151–161, 1968.
- [Lat08] Joachim De Lataillade. Second-order type isomorphisms through game semantics. *Annals of Pure and Applied Logic*, 151(2-3) :115–150, 2008.
- [Lau02] Olivier Laurent. *Etude de la polarisation en logique*. PhD thesis, Université Aix-Marseille II, 2002.
- [Lau04] Olivier Laurent. Polarized games. *Annals of Pure and Applied Logic*, 130(1-3) :79–123, 2004.
- [Lau05] Olivier Laurent. Classical isomorphisms of types. *Mathematical Structures in Computer Science*, 15(5) :969–1004, 2005.
- [Lei04] Tom Leinster. *Higher Operads, Higher Categories*. Cambridge University Press, 2004.
- [Lip02] S. Lippi. *Theorie et pratique des reseaux d'interaction*. PhD thesis, Université de la Méditerranée, 2002.
- [Lor61] Paul Lorenzen. Ein dialogisches Konstruktivitätskriterium. *Infinitistic Methods*, pages 193–200, 1961.
- [LS81] Daniel J. Lehmann and Michael B. Smyth. Algebraic specification of data types : A synthetic approach. *Mathematical Systems Theory*, 14 :97–139, 1981.
- [LS88] J. Lambek and PJ Scott. *Introduction to higher order categorical logic*. Cambridge University Press, 1988.
- [Mat98] R. Matthes. *Extensions of system F by iteration and primitive recursion on monotone inductive types*. PhD thesis, 1998.
- [McC96] Guy McCusker. Games and full abstraction for a functional metalanguage with recursive types. PhD thesis, Imperial College, University of London, 1996. Published in Springer-Verlag's Distinguished Dissertations in Computer Science series, 1998.

- [McC00] Guy McCusker. Games and Full Abstraction for FPC. *Information and Computation*, 160(1-2) :1–61, 2000.
- [Mel95] Paul-André Melliès. Typed lambda-calculi with explicit substitutions may not terminate. In *TLCA*, pages 328–334, 1995.
- [Mel05] Paul-André Melliès. Asynchronous Games 3 : An Innocent Model of Linear Logic. *Electronic Notes in Theoretical Computer Science*, 122 :171–192, 2005.
- [Mel08] Paul-André Melliès. Categorical Semantics of Linear Logic. À paraître dans *Panoramas et Synthèses*, Société Mathématique de France, 2008.
- [Mil77] Robin Milner. Fully abstract models of typed λ -calculi. *Theoretical Computer Science*, 4(1) :1–22, 1977.
- [ML71] Saunders Mac Lane. *Categories for the working mathematician*. Springer, 1971.
- [Mog91] Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93 :55–92, 1991.
- [MP94] Gianfranco Mascari and Marco Pedicini. Head linear reduction and pure proof net extraction. *Theoretical Computer Science*, 135(1) :111–137, 1994.
- [Nic94] Hanno Nickau. Hereditarily sequential functionals. In A. Nerode and Yu. V. Matiyasevich, editors, *Proceedings of the Symposium on Logical Foundations of Computer Science : Logic at St. Petersburg*, volume 813 of *Lecture Notes in Computer Science*, pages 253–264. Springer Verlag, 1994.
- [Plo77] Gordon D. Plotkin. LCF Considered as a Programming Language. *Theoretical Computer Science*, 5(3) :223–255, 1977.
- [Reg] Laurent Regnier. *λ -calcul et réseaux*. PhD thesis.
- [Reg94] Laurent Regnier. Une équivalence sur les lambda-termes. *Theoretical Computer Science*, 126(2) :281–292, 1994.
- [San02a] Luigi Santocanale. Free μ -lattices. *Journal of Pure and Applied Algebra*, 168(2-3) :227–264, 2002.
- [San02b] Luigi Santocanale. μ -bicomplete categories and parity games. *ITA*, 36(2) :195–227, 2002.
- [Sco82] Dana Scott. Domains for denotational semantics. 140 :577–613, 1982.
- [SP82] Michael B. Smyth and Gordon D. Plotkin. The category-theoretic solution of recursive domain equations. *SIAM Journal on Computing*, 11(4) :761–783, 1982.
- [SP00] Alex K. Simpson and Gordon D. Plotkin. Complete axioms for categorical fixed-point operators. In *15th IEEE Symposium on Logic in Computer Science*, pages 30–41, 2000.
- [Tab08] Nicolas Tabareau. *Modalités de ressources et contrôle en logique tensorielle*. PhD thesis, Université Paris Diderot – Paris 7, 2008.

Résumé

LOGIQUE ET INTERACTION : UNE ÉTUDE SÉMANTIQUE DE LA TOTALITÉ

Cette thèse s’articule autour de l’utilisation de stratégies totales pour la représentation des preuves. La première partie porte sur le cadre finitaire. L’analyse commence dans un univers syntaxique : on définit un λ -calcul unaire fortement normalisant, pour lequel on rappelle la machine à pointeurs (PAM). On réduit le problème de préservation de la totalité par composition à un problème de finitude sur des objets appelés structures de pointeurs. On donne trois preuves différentes de ce résultat de finitude. La première se ramène via la PAM à la normalisation du λ -calcul unaire, la seconde passe par l’extraction d’une réduction simple sur les arbres d’entiers et la troisième s’inspire d’un argument combinatoire de Coquand. La seconde partie traite d’un calcul de séquents μLJ équipé de définitions inductives et coinductives, dans lequel on donne une simulation du système T . On définit les catégories μ -fermées, formant une classe de modèles de μLJ . Dans le cadre des jeux on définit les arènes ouvertes, munies de variables de type libres. À chacune de ces arènes ouvertes est associé un foncteur ouvert sur la catégorie des stratégies innocentes. On décrit ensuite sur les arènes ouvertes une construction de boucle dont on montre qu’elle rejoint le modèle de McCusker des types récursifs. Les boucles sont alors enrichies par des conditions de gain inspirées des jeux de parité, ce qui équipe les foncteurs ouverts d’algèbres initiales et coalgèbres terminales et construit une catégorie μ -fermée. On propose finalement une extension de μLJ à une syntaxe infinie, pour laquelle le modèle est pleinement complet.

Mots clés : Sémantique de Jeux – λ -calcul – Théorie de la preuve – Induction – Coinduction

Abstract

LOGIC AND INTERACTION : A SEMANTIC STUDY OF TOTALITY

This thesis deals with the problem of using total strategies for the interpretation of proofs. The first part is about finitary logic. The analysis begins within a syntactical framework : we define a unary strong normalizing λ -calculus, for which we recall the pointer abstract machine (PAM). We reduce the problem of preservation of totality by composition to a finiteness theorem on objects we call pointer structures. We give three different proofs of this result. The first uses a reduction to the normalization of the unary λ -calculus via the PAM, the second extracts from the problem a simple reduction on trees of integers, whereas the third uses a combinatorial argument of Coquand. The second part is a study of a sequent calculus μLJ with inductive and coinductive definitions, in which we give a simulation of Gödel’s system T . We define the notion of μ -closed categories, which are models of μLJ . In the framework of games we define open arenas (arenas with free type variables). To each of these arenas we associate an open functor on the category of arenas and innocent strategies. We then describe on open arenas a loop construction, which we relate to McCusker’s model of recursive types. Loops are enriched with winning conditions imported from parity games, which provide initial algebras and terminal coalgebras to open functors and builds a μ -closed category of games and total innocent strategies. Finally, we give an extension of μLJ to an infinite syntax, for which our model is fully complete.

Keywords : Game semantics – λ -calculus – Proof theory – Induction – Coinduction