



HAL
open science

Un cadre quantitatif pour la Ludique

François Maurel

► **To cite this version:**

François Maurel. Un cadre quantitatif pour la Ludique. Autre [cs.OH]. Université Paris-Diderot - Paris VII, 2004. Français. NNT: . tel-00152476

HAL Id: tel-00152476

<https://theses.hal.science/tel-00152476>

Submitted on 7 Jun 2007

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

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

Thèse présentée et soutenue publiquement
le 26 novembre 2004

en vue de l'obtention du titre de Docteur en Sciences
de l'Université Paris 7 - Denis Diderot
spécialité informatique

Un cadre quantitatif pour la Ludique

François Maurel

Jury

Guy Cousineau	
Pierre-Louis Curien	directeur de thèse
Thomas Ehrhard	rapporteur
Jean-Yves Girard	
J. Martin E. Hyland	rapporteur
Gordon D. Plotkin	

I	Ludique simple	17
1	Notions de ludique simple	19
1.1	Quelques rappels de logique linéaire	20
1.1.1	Règles	21
1.1.2	Focalisation	24
1.1.3	Jeux	25
1.1.4	Un dessein pour commencer	25
1.2	Lieux, actions et desseins	27
1.2.1	Lieux	27
1.2.2	Desseins	27
1.2.3	Réseaux	30
1.2.4	Normalisation	32
1.2.5	Ordres	34
1.3	Théorèmes analytiques	35
1.4	Comportements	37
1.5	Connecteurs	38
1.5.1	Décalages	39
1.5.2	Additifs	39
1.5.3	Multiplicatifs	41
1.6	Quantificateurs	43
1.7	Uniformité	44
1.7.1	Bi-comportements	44
1.7.2	Connecteurs	46
1.8	Séquents de comportements	47
1.9	Complétude	47
1.10	La logique multiplicative additive $MALL_2$	50
1.11	Tranches et ensembles d'actions	51
2	Contributions à la ludique simple	53
2.1	Un problème du tenseur et une solution	54
2.2	Connecteurs	58
2.3	Fax et délocalisation	59
2.4	Ordres	60
2.5	Théorèmes analytiques	60
2.6	Comportements	61
2.7	Tranchettes	61
2.8	Pour la suite	62

II	Ludique probabiliste	63
3	Ludique probabiliste : desseins	65
3.1	Avant la tempête	66
3.2	Objets de base	69
3.2.1	Actions	69
3.2.2	Desseins	74
3.2.3	Réseaux	81
3.3	Normalisation	82
3.3.1	Normalisation par codage	82
3.3.2	Plongement de la ludique simple	95
3.3.3	Normalisation par tranches	97
3.4	Ordres	101
3.5	Théorèmes analytiques	103
3.5.1	Séparation	103
3.5.2	Stabilité	109
3.5.3	Associativité	110
3.5.4	Monotonie	111
4	Ludique probabiliste : à propos d'orthogonalité	113
4.1	Une caractérisation de l'orthogonalité	114
4.1.1	Desseins quasi-simples	115
4.1.2	Couvertures	115
4.1.3	Développement d'un dessin fini	118
4.1.4	Caractérisation	124
4.2	Plongements de la ludique simple	125
5	Ludique probabiliste : comportements et connecteurs	129
5.1	Comportements	130
5.1.1	Définitions	130
5.1.2	Combinaisons linéaires	130
5.1.3	Incarnation	132
5.1.4	Superposition	134
5.1.5	Notions pour la complétude interne des comportements	142
5.2	Connecteurs	145
5.2.1	Décalages	145
5.2.2	Additifs	146
5.2.3	Multiplicatifs	146
5.2.4	Quantificateurs	148
6	Ludique probabiliste : uniformité	149
6.1	Distance	150
6.2	Bi-comportements	152
6.3	Connecteurs	157
6.3.1	Délocalisation et décalage	157
6.3.2	Additifs	158

6.3.3	Multiplicatifs	159
6.4	Complétude vis-à-vis de la ludique simple	160
6.4.1	Décalages	162
6.4.2	Additifs	163
6.4.3	Multiplicatifs	164
6.4.4	Les quantificateurs	164
6.4.5	Séquents de bi-comportements	165
6.5	Complétude forte	165
6.5.1	Condition de gain	165
6.5.2	Correction	166
6.5.3	Complétude	167

III Ludique avec répétitions 169

7	Ludique avec pointeurs	171
7.1	Intuitions	172
7.2	Desseins et réseaux	173
7.2.1	Desseins	173
7.2.2	Réseaux	176
7.3	Normalisation	177
7.3.1	Intuitions	177
7.3.2	Machine abstraite	178
7.3.3	Exemples de normalisations	183
7.4	Ludique avec pointeurs et arbres de Böhm abstraits	185
7.4.1	Arbres de Böhm abstraits	186
7.4.2	Validité des définitions	187
7.5	Plongement de la ludique simple	190
7.6	Ordres	190
7.7	À propos des théorèmes analytiques	191
7.7.1	Non-séparation	191
7.7.2	Associativité	193
7.7.3	Stabilité	193
7.7.4	Monotonie	194
7.8	Comportements	194
7.8.1	Définitions	195
7.8.2	Notions pour la complétude interne	197
7.9	Connecteurs	199
7.9.1	Décalages	199
7.9.2	Exponentielles	200
7.9.3	Additifs	201
7.9.4	Multiplicatifs	202

8	Ludique exponentielle	205
8.1	Intuitions	206
8.2	Desseins et réseaux	207
	8.2.1 Dessains	208
	8.2.2 Réseaux	209
8.3	Normalisation	210
	8.3.1 Traduction en ludique avec pointeurs	210
	8.3.2 Plongement de la ludique avec pointeurs	214
8.4	Ordres	215
8.5	Théorèmes analytiques	216
	8.5.1 Associativité	216
	8.5.2 Séparation	217
	8.5.3 Stabilité	225
	8.5.4 Monotonie	226
8.6	Comportements	227
	8.6.1 Définitions	227
	8.6.2 Notions pour la complétude interne	229
8.7	Connecteurs	230
	8.7.1 Décalages	230
	8.7.2 Exponentielles	231
	8.7.3 Additifs	232
	8.7.4 Multiplicatifs	232
8.8	Uniformité	233
	8.8.1 Bi-comportements	233
	8.8.2 Connecteurs	235
	8.8.3 Décalages	235
	8.8.4 Exponentielles	236
	8.8.5 Additifs	237
	8.8.6 Multiplicatifs	237
8.9	Un mot sur l'interprétation	239

Remerciements

Je tiens tout d'abord à remercier Pierre-Louis Curien pour m'avoir accueilli à PPS. Il m'a toujours patiemment encouragé. Au cours de cette thèse, j'ai souvent découvert qu'il avait déjà travaillé sur une idée ou une autre quelques années auparavant.

Je remercie collectivement mon jury pour avoir accepté d'en faire partie malgré des emplois du temps chargés. Je suis très honoré car chaque membre de ce jury a participé par ses travaux de façon déterminante à certains de mes centres d'intérêt scientifiques présentés ou non dans ce mémoire. Je remercie grandement Thomas Ehrhard et Martin Hyland qui ont accepté la lourde charge de rapporteurs. Je remercie également Jean-Yves Girard qui a accepté de me voir à plusieurs reprises durant cette thèse pour me guider, questionner mes choix et pour me relancer au moment opportun. Enfin, je remercie chaleureusement Guy Cousineau et Gordon Plotkin qui ont bien voulu compléter ce jury et apporter un regard extérieur au domaine.

J'aimerais aussi exprimer ma reconnaissance à Antonio Bucciarelli, Vincent Danos, Paul-André Melliès et Vincent Padovani ainsi que plus globalement au reste de l'équipe PPS à la fois pour leurs qualités humaines, scientifiques et pédagogiques mais également pour l'effervescence qui caractérisait le laboratoire quand j'ai eu la chance de le rejoindre.

Je voudrais aussi saluer mes co-auteurs Sylvain Baro, Claudia Faggian et Emmanuel Belfara avec lesquels j'ai eu grand plaisir à travailler ainsi que quelques autres personnages que j'ai pu côtoyer pendant ces années de thèse : Odile Ainardi, des co-bureaux exceptionnels : Michel Hirschowitz, Raphaël Montelatici, Feng Yangyue, Deng Yuxin, Fabien Tarissan, Zhang Min, Samuel Hym, quelques thésards d'en-face presque aussi exceptionnels : Vincent Balat, Jean-Vincent Lodo, Emmanuel Polonovski, Anne-Gwenn Bosser, Benjamin Leperchey, Joachim de Lataillade ainsi que les courageux thésards du laboratoire d'à côté : Denis Oddoux et Pierre Moro qui ont même osé venir nous voir. Ils ont tous contribué à faire de ma thèse un moment agréable et certains ont même accepté quelques relectures ou discussions au moment opportun.

Je n'oublie pas des amis thésards de plus longue date avec qui j'ai eu la chance de faire des études, d'en parler, de ne pas en parler, . . . : Matthieu Finiasz, Bénédicte Py, Vincent Simonet. . .

Enfin, je remercie Marianne Durand sans qui cette thèse et surtout le reste seraient aussi ternes que différents.

Introduction

La logique de la programmation

La *logique de la programmation* est le domaine à la frontière de l'informatique et de la logique qui s'intéresse aux preuves mathématiques suivant l'optique de la correspondance de Curry-Howard. Ce résultat stipule qu'une preuve d'une formule A est un programme, c'est-à-dire une fonction, de type A . L'intérêt principal d'un programme de type A réside généralement plus dans sa *façon* de calculer que dans son type. De même, dans le cadre de Curry-Howard, une preuve d'une formule A va souvent être plus considérée pour la manière dont elle prouve A que pour ce qu'elle prouve.

Étant donnée une entrée x , un programme f calcule la sortie $f(x)$ en plusieurs étapes de calcul. De même, étant donnée une preuve x d'une formule A , une preuve f de $A \rightarrow B$ calcule une preuve de B en "recollant" la preuve x sur le bout de preuve qui correspond à A dans f (c'est ce qu'on appelle une *coupure*) puis en réduisant la preuve obtenue (c'est la phase d'élimination des coupures).

La correspondance de Curry-Howard fut originellement remarquée pour le λ -calcul simplement typé, le noyau des langages fonctionnels, mais a été étendue depuis à des logiques de plus en plus expressives.

La logique linéaire

La logique classique est la logique qu'on emploie, généralement, pour faire des raisonnements. La logique intuitionniste est fondée sur une restriction des règles de la logique classique de telle manière que les preuves correspondent naturellement à des fonctions. C'est dans ce cadre que la correspondance de Curry-Howard est la plus facile à obtenir même si elle a été étendue depuis à toute la logique classique.

La logique linéaire [Gir95b], introduite par Jean-Yves Girard en 1986, est une logique de *ressources* en ce sens qu'elle permet de décomposer la logique intuitionniste (plus exactement l'implication intuitionniste) en une partie "exponentielle" qui s'occupe des duplications et effacements des arguments d'un programme et une partie "linéaire", qui se contente de manipuler des preuves sans événements coûteux en mémoire. La formule habituelle qui résume cette décomposition est

$$A \rightarrow B = !A \multimap B$$

Elle signifie qu'une fonction intuitionniste de A dans B est une fonction linéaire (comme indiqué dans la flèche linéaire \multimap) qui prend des copies de A (comme indiqué par $!A$)

et renvoie un exemplaire de B . Cette explication utilise naturellement le dictionnaire de Curry-Howard : on parle indépendamment de fonction ou de preuve, de type ou de formule.

Il y a deux connecteurs exponentiels en logique linéaire : le *bien sûr*, noté $!$, que l'on a déjà évoqué et son dual, le *pourquoi pas*, noté $?$. Ils servent à créer, dupliquer et effacer des copies de manière explicite. Ces règles logiques remplacent les règles *structurelles* implicites de la logique habituelle.

Les autres connecteurs principaux sont le *tenseur* (noté \otimes), le *par* (noté \wp), le *avec* (noté $\&$) et le *plus* (noté \oplus). Le *tenseur* et le *avec* sont deux formes de “et” qui sont distinguées grâce au caractère explicite des exponentielles car les règles structurelles ne s'appliquent pas sur des formules quelconques.

D'un point de vue intuitif de ressource, la distinction entre le *tenseur* et le *avec* est claire : si de A on peut obtenir B et que de A on peut obtenir C alors de A on peut obtenir $B\&C$ mais on ne peut pas forcément obtenir $B\otimes C$. Par contre, de $A\otimes A$, on peut obtenir $B\otimes C$. Ces remarques sont résumées dans ce qui est communément appelé le *menu gastronomique* de Lafont. Le menu peut être décrit comme suit. Dans un restaurant, il y a un menu à 15 euros qui propose

<p>Entrée : tomate pâté</p> <p>Plat : grillade poisson frais ou pané (suivant arrivage)</p> <p>Dessert : buffet de fruits à volonté</p>
--

Le consommateur s'attend à manger de la tomate ou du pâté puis une grillade ou du poisson (il ne choisit pas la qualité de son poisson) et enfin autant de fruits qu'il désire. Tout cela coûte 15 euros. Un couple paie, naturellement, 30 euros pour ce repas. Cette description évidente à tout gourmet peut être “formalisée” en la formule suivante

$$15 \text{ eur} \multimap ((\text{tomate} \& \text{pâté}) \otimes (\text{grillade} \& (\text{poisson frais} \oplus \text{poisson pané}))) \otimes ! \text{fruit}$$

Les intuitions dégagées par cet exemple sont utiles pour la compréhension de la logique linéaire mais ne sauraient remplacer une étude plus formelle.

Un dernier point intéressant de la logique linéaire est la négation qui est définie par les règles de De Morgan et non comme un connecteur sur les formules. Elle s'apparente à la négation de la logique classique qui peut ainsi être plongée en logique linéaire.

Les sémantiques qualitatives et quantitatives

Les modèles de calcul peuvent être séparés en deux familles distinctes : les modèles qualitatifs et les modèles quantitatifs. Les modèles qualitatifs comme les espaces de cohé-

rence [Gir95b, Gir99] ne prennent généralement pas en compte le nombre de copies d'une donnée alors que les modèles quantitatifs essaient d'en tirer partie.

Les exemples les plus connus de modèles quantitatifs dans le cadre de la logique linéaire sont les *espaces de Banach cohérents* introduits par Jean-Yves Girard [Gir96a], une extension des espaces de cohérence par Nuno Barreiro et Thomas Ehrhard [BE99] et les *espaces de Köthe* introduits par Thomas Ehrhard [Ehr01, Ehr04].

Les polarités

La notion de polarité utilisée ici provient d'une étude sur la forme des règles de la logique linéaire. Les formules sont divisées en deux classes : les formules positives et les formules négatives. Les connecteurs sont aussi polarisés : on dénombre les connecteurs positifs (qui prennent des formules positives et renvoient une formule positive), les connecteurs négatifs (qui prennent des formules négatives et renvoient une formule négative) et des connecteurs de changement de polarité qui sont typiquement les connecteurs exponentiels.

Tout ceci est résumé dans la grammaire des formules polarisées de la logique linéaire :

$$\begin{aligned} P &:= X \mid P \otimes P \mid P \oplus P \mid 1 \mid 0 \mid \exists X.P \mid !N \\ N &:= X^\perp \mid N \wp N \mid N \& N \mid \perp \mid \top \mid \forall X.N \mid ?P \end{aligned}$$

L'intuition habituelle est qu'une formule positive envoie une information tandis qu'une formule négative attend une information.

Ainsi, une preuve de $A \& B$ est une paire composée d'une preuve π_A de A et d'une preuve π_B de B ; une preuve de $A^\perp \oplus B^\perp$ est soit une preuve de A^\perp , soit une preuve de B^\perp .

Une coupure entre $A \& B$ et $A^\perp \oplus B^\perp$ s'élimine de la façon suivante. Si la preuve de $A^\perp \oplus B^\perp$ est une preuve π_{A^\perp} de A^\perp alors on obtient une coupure entre π_A et π_{A^\perp} . Symétriquement, si la preuve de $A^\perp \oplus B^\perp$ est une preuve π_{B^\perp} de B^\perp alors on obtient une coupure entre π_B et π_{B^\perp} . La preuve de $A \& B$ a "attendu" que la preuve de $A^\perp \oplus B^\perp$ lui envoie l'information nécessaire (π_{A^\perp} ou π_{B^\perp}).

Une étude générale de la notion de polarité a été menée par Olivier Laurent dans sa thèse [Lau02].

Les jeux

La sémantique de jeux provient de l'idée suivante : pour comprendre le calcul fait par un programme dans un environnement ou entre deux programmes qui échangent des données, il suffit de regarder l'ensemble des *interactions* entre le programme qui nous intéresse et le monde extérieur. Ces traces de calcul (les interactions) forment la sémantique de jeux du programme.

Les travaux précurseurs aux jeux sont les algorithmes séquentiels [BC82] de Gérard Berry et Pierre-Louis Curien où l'on trouve l'idée d'interactions primitives qui alternent entre un programme et son environnement sous forme de questions et de réponses.

De grands succès de la sémantique de jeux sont la caractérisation de traits de langages. Ainsi, les travaux de 1994 sur PCF [HO00, AJM00] de Martin Hyland et Luke Ong d’une part et de Samson Abramsky, Radha Jagadeesan et Pasquale Malacaria d’autre part ont popularisé les jeux comme sémantique commune de plusieurs langages. Les différences entre ces langages sont descriptibles par la géométrie particulière des jeux retenus pour représenter ces langages.

Les travaux sur les jeux pour la logique linéaire sont nombreux. On peut citer les travaux de Samson Abramsky, Patrick Baillot, Vincent Danos, Thomas Ehrhard, Radha Jagadeesan, Hugo Herbelin, Olivier Laurent, Paul-André Melliès et Laurent Regnier [AJ94, DHR96, BDER97a, BDER97b, AM99, Lau04].

Les sémantiques de jeux ne se cantonnent naturellement pas à la logique linéaire. Pour cette thèse, des jeux importants sont les arbres de Böhm abstraits [Cur98] de Pierre-Louis Curien qui apportent une sémantique de jeux de divers formalismes autour de PCF à la manière des jeux dits de Hyland et Ong. Ils sont importants dans notre cadre car les desseins de la ludique sont des sortes d’arbres de Böhm abstraits [Cur04b].

La séparation

La séparation est une notion fondamentale en ludique. On dit qu’un ensemble de joueurs est *séparé* si pour toute paire de joueurs dans cet ensemble, il existe un autre joueur qui n’interagit pas de la même manière avec chacun d’eux. On dit que cet autre joueur les *sépare*.

Si on appelle sémantique d’un joueur la manière dont il joue, la séparation stipule donc que la syntaxe correspond exactement à la sémantique. Ainsi, deux joueurs différents jouent de manière différente.

Cette propriété pourrait sembler anecdotique mais a des conséquences importantes comme dans la preuve du théorème de complétude en ludique.

La ludique

La ludique [Gir01] introduite par Jean-Yves Girard en 2001, est une sémantique de jeux [FH02] de la logique linéaire qui possède des propriétés de structure très importantes.

Elle est fondée sur quelques briques de base qui ont pour but de *géométriser* le calcul entre un joueur et son opposant. Ce calcul doit être le plus symétrique possible et en particulier, joueur et opposant doivent vérifier les mêmes propriétés, à la polarité près. Les types doivent apparaître naturellement. Ainsi, un joueur et son opposant peuvent “se mettre d’accord” dans un sens ou un autre (en ludique, on utilise le démon comme expliqué dans la suite). On dit qu’ils sont alors orthogonaux. Un type, appelé en ludique un comportement, est l’ensemble des joueurs orthogonaux à un ensemble donné d’opposants. En général, l’orthogonalité peut être définie de plusieurs façons. La méthode est ici d’utiliser un objet particulier : le *démon*. Cet objet peut être compris comme une exception non rattrapée dans un programme [Cur04b]. Un joueur et son opposant sont orthogonaux si l’un des deux joue le démon pendant leur interaction. Le dernier ingrédient est la notion de *lieu*. Les formules de la logique linéaire sont remplacées par leur “position” dans

l'espace : leur lieu. Ce changement conceptuel apporte de nouveaux phénomènes comme l'existence assez générale de formes prenexes.

Une fois ces briques de base bien définies, la construction de la ludique se fait en plusieurs étapes : (1) les joueurs (appelés desseins) sont des arbres essentiellement linéaires ; (2) la normalisation, qui correspond à l'élimination des coupures, se fait comme en logique linéaire multiplicative additive (sans les exponentielles) ; (3) on prouve des théorèmes structurels (les théorèmes *analytiques* : séparation, associativité...) ; (4) les comportements sont définis, (5) une condition de gain sur les desseins est définie puis (6) on atteint le théorème de complétude (toute preuve correspond à un dessein *gagnant* et tout dessein *gagnant* à une preuve). L'intuition pour les théorèmes analytiques est qu'ils suffisent pratiquement pour prouver le théorème de complétude (il ne manque essentiellement qu'une notion d'*uniformité* pour la condition de gain).

Malgré ses qualités exceptionnelles, la ludique peut être améliorée dans plusieurs directions :

Exponentielles. La ludique est un cadre linéaire dans le sens qu'on ne peut répéter deux fois le même coup. Il manque donc des objets pour représenter les répétitions ce qui permettrait de chercher un théorème de complétude pour la logique linéaire en entier.

Divergence. La notion de divergence, qui correspond à un calcul mal terminé, n'est pas si bien comprise et quelques corrections sont possibles. En particulier, la ludique identifie le fait de ne pas terminer par interaction infinie ou parce qu'il manque une donnée nécessaire à la suite du calcul.

Séquentialité. La ludique est un cadre très séquentiel ; l'interaction est complètement déterministe. Il peut être intéressant de voir si l'on peut construire un modèle moins séquentiel pour se rapprocher des réseaux de preuves (une syntaxe parallèle pour la logique linéaire [Gir96c]).

Probabilité. Dans le même genre d'idée, plutôt que de rajouter du parallélisme, on pourrait rajouter des probabilités comme cela a été fait pour les jeux par Vincent Danos et Russel Harmer [DH02]. Il faut tout de même garder à l'esprit qu'il ne faut entreprendre cela que pour gagner de nouvelles propriétés ou comprendre le modèle actuel et non simplement pour avoir un modèle probabiliste.

Non-déterminisme. On pourrait aussi chercher une variante non-déterministe de la ludique dans l'esprit des jeux non-déterministes [Har99].

Mis-à-part le point sur la divergence qui est une conséquence de ce travail, ces directions étaient plus ou moins claires au début de cette thèse. La direction la plus importante, mais aussi la plus floue a priori, est naturellement la question des exponentielles.

Cette thèse

Ce mémoire se situe dans la continuité de la ludique et vise à compléter les constructions ludiques pour étendre son champ d'application. Les motivations principales pour cette thèse sont à trouver dans le constat des améliorations possibles listées précédemment.

L'évidence de l'intérêt d'un modèle **probabiliste** de la ludique provient de l'intuition suivante. En présence de **répétitions**, les probabilités permettent de "compter" le nombre de coups identiques à une action donnée. On devrait ainsi pouvoir séparer un joueur qui joue quelques coups puis joue le démon d'un joueur qui joue ces mêmes coups puis les répète à l'identique et enfin joue le démon. Accessoirement, la ludique probabiliste donne des pistes de compréhension sur la divergence en ludique simple. En particulier, elle *différencie* la divergence en temps fini de la divergence par interaction infinie.

Les questions de **séquentialité** ne sont pas abordées ici. Un travail actuel avec Claudia Faggian tente d'y répondre. Les questions de **non-déterminisme** ne sont pas non plus abordées car elles n'ont pas mené à des résultats intéressants.

Le mémoire qui suit est composé de trois parties.

La **première partie** est consacrée à la ludique telle que définie par Jean-Yves Girard que l'on appelle la **ludique simple** pour la différencier des variantes possibles. Elle se compose de deux chapitres : un chapitre de rappel qui pourra servir de référence dans la suite du texte et un chapitre corrigeant quelques détails de la construction. Ce deuxième chapitre introduit les notions de *divergence négative* et de *démon focalisé* qui sont utiles pour l'adjonction du *tenseur*. Ces deux notions ont été introduites après les premières constructions en ludique probabiliste car elles deviennent indispensables dans ce cadre. Cependant, le chapitre de rappel ne se contente pas uniquement de rappeler les définitions et théorèmes. La définition de *dessin* est totalement évacuée, renforçant au passage le point de vue géométrique. Cela a, entre autres choses, impliqué de caractériser la notion de dessein exact sans *dessin*. Cette partie présente aussi un élément du folklore de la ludique que sont les tranches déséquentialisées (des morceaux de preuve dans lequel les informations de séquentialité sont oubliées). L'intérêt de cette notion est justifié dans cette thèse par la notion de divergence négative.

La **deuxième partie** est consacrée à la construction de la **ludique probabiliste** linéaire. Elle suit essentiellement le plan du chapitre 1 : desseins, normalisation, comportements, connecteurs, uniformité et complétude. Néanmoins, plusieurs ingrédients sont originaux :

1. La notion de coefficient est "libre" dans le sens qu'ils sont construits librement autour des idées de scalaires (dans $[0; 1]$), de somme finie, et de séquence. Cette construction, qui n'utilise pas simplement des mesures, sert pour la séparation et évite quelques passages à la limite dans les preuves. Incidemment, cette géométrie ouvre la voie à la méthode de définition de la normalisation.
2. La normalisation est définie par un codage en ludique simple qui consiste à rajouter un lieu dans chaque dessein pour coder les coefficients. Cette technique simplifie les preuves et est réemployée en diverses occasions. En particulier, un résultat de conservation d'*équations génériques* montre que des équations d'une certaine sorte sont vraies en ludique simple si et seulement si elles sont vraies en ludique probabiliste. Cette méthode donne une preuve immédiate de l'associativité de la ludique probabiliste. Elle est généralisée pour montrer le théorème de monotonie.
3. La preuve de séparation est menée grâce à l'introduction de desseins *universels* : des desseins probabilistes qui contiennent, en un certain sens, tous les desseins simples

de même base. Étant donnés deux desseins distincts, il est possible de construire deux desseins universels dont au moins un des deux les séparent. Un résultat plus général est prouvé : toute paire de desseins universels distincts sépare les desseins sur la base duale.

4. Pour la notion d'uniformité, une équivalence partielle sur les desseins en ludique simple, la technique employée a consisté à utiliser une "distance". Un dessin uniforme, c'est-à-dire un candidat à représenter une preuve, est un dessin à "distance" nulle de lui-même.
5. Le théorème de complétude forte en ludique probabiliste est d'abord montré vis-à-vis de la ludique simple. Cette méthodologie a deux conséquences principales : (1) on utilise les résultats de la ludique simple au maximum et on n'adapte ces résultats que lorsque c'est nécessaire ; et (2) par composition des résultats de complétude forte entre la ludique simple et la logique linéaire et entre la ludique probabiliste et la ludique simple, on obtient des résultats de complétude forte pour les deux systèmes multiplicatifs/additifs $MALL_2$ et $MAAL_2$.

La **troisième partie** est consacrée à l'ajout de répétitions et aux **exponentielles**.

1. Le premier chapitre est consacré à la ludique avec *pointeurs*. L'idée directrice est d'ajouter des pointeurs à la ludique simple pour gérer les répétitions. Pour la normalisation, une machine abstraite est définie à partir de la SVAM (Strong View Abstract Machine) [Cur98]. Une première interprétation des connecteurs exponentiels en ludique est donnée dans ce chapitre.
Parmi les théorèmes analytiques, seule la séparation n'est pas vérifiée.
2. Ce point est corrigé dans le deuxième et dernier chapitre qui est consacré à la ludique exponentielle : une fusion de la ludique avec pointeurs et de la ludique probabiliste. Les théorèmes analytiques sont prouvés en utilisant les techniques et résultats des chapitres précédents.
3. Comme en ludique probabiliste, la séparation utilise une notion de dessin universel mais celle-ci est plus compliquée car chaque dessin avec pointeurs est "inclus" une infinité de fois dans un dessin universel de sa base. Le résultat de séparation utilisant deux desseins fixés se conserve.
4. Les définitions des connecteurs de la ludique avec pointeurs sont généralisées dans ce cadre. Enfin, une notion d'uniformité dans l'esprit de l'uniformité en ludique probabiliste est ajoutée même si aucun résultat de complétude n'est avancé.

Le concept de lieu est peut-être le plus important de toute la rhétorique. Définissable, en théorie actuelle des figures, comme figure macrostructurale de second niveau, le lieu peut être appréhendé, très généralement, comme un stéréotype logico-discursif. C'est la base essentielle des preuves techniques de l'argumentation et la matière de l'invention. Selon la Logique de Port-Royal, les lieux sont certains chefs généraux auxquels on peut rapporter toutes les preuves dont on se sert dans les diverses matières que l'on traite. L'idée de lieu, élaborée par Aristote, relève donc d'une pensée scientifique extrêmement puissante et absolument moderne, qui radicalise le langage et la logique en construisant en outre une combinaison structurale particulièrement complexe de toutes les formes pensables du raisonnement argumentatif naturel, appuyée sur un effort d'abstraction et de généralisation assez extraordinaire. (...)

G. Moulinié, Dictionnaire de rhétorique

Première partie

Ludique simple

Chapitre 1

Notions de ludique simple

Sommaire

1.1	Quelques rappels de logique linéaire	20
1.1.1	Règles	21
1.1.2	Focalisation	24
1.1.3	Jeux	25
1.1.4	Un dessein pour commencer	25
1.2	Lieux, actions et desseins	27
1.2.1	Lieux	27
1.2.2	Desseins	27
1.2.3	Réseaux	30
1.2.4	Normalisation	32
1.2.5	Ordres	34
1.3	Théorèmes analytiques	35
1.4	Comportements	37
1.5	Connecteurs	38
1.5.1	Décalages	39
1.5.2	Additifs	39
1.5.3	Multiplicatifs	41
1.6	Quantificateurs	43
1.7	Uniformité	44
1.7.1	Bi-comportements	44
1.7.2	Connecteurs	46
1.8	Séquents de comportements	47
1.9	Complétude	47
1.10	La logique multiplicative additive MALL_2	50
1.11	Tranches et ensembles d'actions	51

L'objet de ce chapitre est tout d'abord de rappeler les définitions de base de la ludique, introduite par Jean-Yves Girard [Gir01]. Ce chapitre résume et rappelle les nombreuses notions, et leur vocabulaire, de la ludique simple. L'objectif est de rappeler et parfois de motiver les définitions et propositions qui servent par la suite sans prétendre enseigner la ludique ou introduire ses notions. En un mot, ce chapitre sert de référence dans cette thèse pour le domaine au vocabulaire riche qu'est la ludique.

La ludique standard, comme définie par Jean-Yves Girard, est ici qualifiée de simple ainsi que les notions qui s'y rapportent. Le terme de *simple* s'oppose à *probabiliste*, *multiple*, *exponentiel* ou *concurrent*...

La première section rappelle brièvement des notions de logique linéaire et donne des clefs pour la compréhension des polarités et de la focalisation. Les sections suivantes définissent, en suivant généralement le plan de l'article fondateur [Gir01], les notions fondamentales de la ludique : lieux, démon, desseins, théorèmes principaux, comportements, connecteurs, uniformité et complétude.

Quelques sources pour la section sur la logique linéaire sont les articles de Jean-Yves Girard [Gir99, Gir95b], Olivier Laurent [Lau02] et l'introduction de Pierre-Louis Curien [Cur04a, Cur04b]. Quelques sources pour comprendre la suite du chapitre sont les articles de Jean-Yves Girard [Gir98, Gir01], la thèse de Claudia Faggian [Fag02], l'introduction de Pierre-Louis Curien [Cur04b] et naturellement les travaux sur les jeux [Lau02, Lau04, FH02, BDER97a, HO00, AJM00]. Pour la notion d'uniformité, on pourra aussi se reporter à une introduction [FFDQ02].

1.1 Quelques rappels de logique linéaire

Cette section rappelle les premières définitions de la logique linéaire [Gir95b] et revient sur les notions fondamentales de polarité et de focalisation qui prévalent en ludique.

La logique linéaire a été introduite par Jean-Yves Girard en 1986. Les liens avec l'informatique sont immédiats car elle apporte une notion de gestion de ressources. Les connecteurs sont divisés en deux parties : les multiplicatifs (le *tenseur* \otimes et le *par* \wp) et les additifs (le *avec* $\&$ et le *plus* \oplus). Un grand succès en est la négation involutive définie par les lois de De Morgan. Il y a de plus deux modalités (dites exponentielles) pour gérer la duplication et l'effacement de ressources (le *bien sûr* $!$ et le *pourquoi pas* $?$).

La séparation multiplicatif/additif n'est pas la seule possible. On peut aussi classer les formules suivant une polarité positif/négatif. Commençons par étudier les règles.

Définition 1.1 – Formules de la logique linéaire

Les **formules de la logique linéaire** sont données par la grammaire :

$$\begin{aligned}
 A & := X \mid A \otimes A \mid A \oplus A \mid 1 \mid 0 \mid \exists X.A \mid !A \\
 & \mid X^\perp \mid A \wp A \mid A \& A \mid \perp \mid \top \mid \forall X.A \mid ?A
 \end{aligned}$$

Définition 1.2 – Négation linéaire

Pour toute formule A , sa **négation**, notée A^\perp , est définie comme son dual par les lois de De Morgan (avec $A^{\perp\perp} = A$)

$$\begin{array}{lll} (X)^\perp = X^\perp & 1^\perp = \perp & 0^\perp = \top \\ (A \otimes B)^\perp = (A^\perp \wp B^\perp) & (A \oplus B)^\perp = (A^\perp \& B^\perp) & \\ (!A)^\perp = ?(A^\perp) & (\exists X.A)^\perp = \forall X.(A^\perp) & \end{array}$$

1.1.1 Règles

Les règles de la logique linéaire sont données figure 1.1 page suivante. Les séquents sont de la forme $\vdash \Gamma$ où Γ est un multi-ensemble de formules. La règle d'échange est considérée comme automatique : si Γ' est une permutation de Γ , alors on a la règle $\frac{\vdash \Gamma}{\vdash \Gamma'}$.

Une remarque immédiate, et partiellement reflétée dans la graphie des connecteurs, est que les règles pour le *par*, le *avec* et le *pour tout* ont la propriété que les prémisses peuvent se déduire de la conclusion ce qui n'est pas le cas du *tenseur*, du *plus* ou du *il existe*. Cela amène à la notion de réversibilité.

Définition 1.3 – Règle réversible

Une règle $\frac{\vdash \Gamma_1 \quad \dots \quad \vdash \Gamma_n}{\vdash \Gamma}$ est **réversible** lorsque l'existence d'une preuve π du séquent $\vdash \Gamma$ implique celles de preuves des séquents $\vdash \Gamma_i$.

Les règles pour le *par*, le *avec* et le *pour tout* sont réversibles et ces connecteurs sont appelés négatifs. Dualelement, les connecteurs *tenseur*, *plus* et *il existe* sont appelés positifs.

Ces connecteurs vérifient aussi une propriété de structure : la focalisation. Un arbre de connecteurs positifs se comporte comme un connecteur synthétique [And90], un *connecteur* qui regroupe plusieurs connecteurs de base.

Cette reconnaissance de ces familles positifs/négatifs pousse à restreindre les formules aux formules polarisées (un connecteur d'une certaine polarité s'applique à des formules de cette même polarité) et à utiliser les exponentielles pour changer de polarité.

Les formules polarisées de la logique linéaire sont données par la grammaire :

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

Dans le cas sans exponentielles, il faut des connecteurs pour changer de polarités. Ce sont les décalages, qui peuvent être vus comme le pendant linéaire (ou affine) des connecteurs exponentiels. Les formules polarisées de la logique linéaire multiplicative additive sont données par la grammaire :

$$\begin{array}{l} P := X \mid P \otimes P \mid P \oplus P \mid 1 \mid 0 \mid \exists X.P \mid \downarrow N \\ N := X^\perp \mid N \wp N \mid N \& N \mid \perp \mid \top \mid \forall X.N \mid \uparrow P \end{array}$$

Les règles de déduction de la logique linéaire multiplicative additive sont données figure 1.2 page 23.

axiome et coupure	$\frac{}{\vdash X, X^\perp}$	$\frac{\vdash \Gamma, A \quad \vdash A^\perp, \Delta}{\vdash \Gamma, \Delta}$
multiplicatifs	$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B}$	$\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \wp B}$
additifs	$\frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B}$	$\frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \& B}$
constantes	$\frac{}{\vdash 1}$	$\frac{\vdash \Gamma}{\vdash \Gamma, \perp}$
exponentielles	$\frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A}$	$\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}$
quantificateurs	$\frac{\vdash \Gamma, A[B/X]}{\vdash \Gamma, \exists X.A}$	$\frac{\vdash \Gamma, A}{\vdash \Gamma, \forall X.A} \quad X \notin \text{Fv}(\Gamma)$

FIG. 1.1 – Règles de la logique linéaire

axiome et coupure	$\frac{}{\vdash X, X^\perp}$	$\frac{\vdash \Gamma, A \quad \vdash A^\perp, \Delta}{\vdash \Gamma, \Delta}$
multiplicatifs	$\frac{\vdash \Gamma, P \quad \vdash \Delta, Q}{\vdash \Gamma, \Delta, P \otimes Q}$	$\frac{\vdash \Gamma, N, M}{\vdash \Gamma, N \wp M}$
additifs	$\frac{\vdash \Gamma, P}{\vdash \Gamma, P \oplus Q}$	$\frac{\vdash \Gamma, N \quad \vdash \Gamma, M}{\vdash \Gamma, N \& M}$
constantes	$\frac{}{\vdash 1}$	$\frac{\vdash \Gamma}{\vdash \Gamma, \perp}$
		$\frac{}{\vdash \Gamma, \top}$
quantificateurs	$\frac{\vdash \Gamma, P[A/X]}{\vdash \Gamma, \exists X.P}$	$\frac{\vdash \Gamma, N}{\vdash \Gamma, \forall X.N} \quad X \notin \text{Fv}(\Gamma)$
décalages	$\frac{\vdash \Gamma, N}{\vdash \Gamma, \downarrow N}$	$\frac{\vdash \Gamma, P}{\vdash \Gamma, \uparrow P}$

FIG. 1.2 – Règles de la logique multiplicative additive avec décalages

La ludique [Gir01] se place dans ce cadre multiplicatif additif avec des décalages pour changer de polarité. Dans la logique linéaire polarisée, Olivier Laurent limite les séquents à ne contenir qu'un positif au plus. Cette contrainte permet de généraliser les règles structurales (contraction et affaiblissement) à toutes les formules négatives et non seulement aux formules $?A$. Pour le théorème de complétude de la ludique, Jean-Yves Girard utilise une condition similaire à l'aide d'un *bénitier* : une seule formule positive est utilisée à un moment donné. La notion de bénitier apparaît déjà dans LC [Gir91a], une version constructive de la logique classique. Elle n'est pas sans rapport avec les séquents triadiques de Jean-Marc Andreoli [And90].

1.1.2 Focalisation

La reconnaissance de l'importance de la dualité positif/négatif due à Jean-Marc Andreoli [And90] permet de considérer une logique avec des connecteurs synthétiques [Gir99], c'est-à-dire des règles de la forme

$$\frac{\vdash \Gamma_1, N_1 \quad \dots \quad \vdash \Gamma_n, N_n}{\vdash \Gamma_1, \dots, \Gamma_n, \phi(\downarrow N_{i_1}, \dots, \downarrow N_{i_k})}$$

où ϕ est un arbre de connecteurs positifs *tenseurs* et *plus*.

Par exemple, le fragment de preuve suivant :

$$\frac{\frac{\frac{\vdash \Gamma_1, N_1}{\vdash \Gamma_1, \downarrow N_1} \quad \frac{\vdash \Gamma_2, N_2}{\vdash \Gamma_1, \downarrow N_2}}{\vdash \Gamma_1, \Gamma_2, \downarrow N_1 \otimes \downarrow N_2} \quad \frac{\vdash \Gamma_3, N_3}{\vdash \Gamma_3, \downarrow N_3 \oplus \downarrow N_4}}{\vdash \Gamma_1, \Gamma_2, \Gamma_3, (\downarrow N_1 \otimes \downarrow N_2) \otimes (\downarrow N_3 \oplus \downarrow N_4)}$$

peut-être "compressé" en la règle synthétique obtenue en enlevant toutes les règles intermédiaires :

$$\frac{\vdash \Gamma_1, N_1 \quad \vdash \Gamma_2, N_2 \quad \vdash \Gamma_3, N_3}{\vdash \Gamma_1, \Gamma_2, \Gamma_3, (\downarrow N_1 \otimes \downarrow N_2) \otimes (\downarrow N_3 \oplus \downarrow N_4)}$$

Pour le connecteur synthétique défini ainsi, il y a une autre règle droite :

$$\frac{\vdash \Gamma_1, N_1 \quad \vdash \Gamma_2, N_2 \quad \vdash \Gamma_4, N_4}{\vdash \Gamma_1, \Gamma_2, \Gamma_4, (\downarrow N_1 \otimes \downarrow N_2) \otimes (\downarrow N_3 \oplus \downarrow N_4)}$$

On aura symétriquement des connecteurs synthétiques négatifs.

La construction de connecteurs synthétiques négatifs ou positifs ne peut pas être généralisée à toutes les formules ϕ . Jean-Yves Girard [Gir99] donne l'exemple de $\phi(A, B, C) = A \oplus (B \& C)$. Montrons le pour $\phi(A, B, C) = A \otimes (B \wp C)$. Supposons que l'on possède un ensemble de règles complet concluant ϕ et un autre pour ϕ^\perp , le connecteur dual.

On essaie de prouver le séquent $\vdash P^\perp \otimes (Q^\perp \wp R^\perp), P \wp (Q \otimes R)$ sans utiliser d'axiome immédiatement. La première règle utilisée ne peut décomposer le terme de gauche car il faudrait partager la formule de droite. Elle doit donc partager le terme de droite. Elle "ouvre" le *par* puis doit alors réduire le *tenseur* de droite *avant* celui de gauche. C'est impossible pour des raisons de contexte ce qui implique que ϕ n'est pas un connecteur. La seule preuve de $\phi(P, Q, R)$ autre que l'axiome commence par

$$\frac{\frac{\overline{\vdash P^\perp, P} \quad \overline{\vdash Q^\perp \wp R^\perp, Q \otimes R}}{\vdash P^\perp \otimes (Q^\perp \wp R^\perp), P, Q \otimes R}}{\vdash P^\perp \otimes (Q^\perp \wp R^\perp), P \wp (Q \otimes R)}$$

On y remarque clairement l'alternance droite/gauche (le *par* de droite puis le *tenseur* de gauche) alors que pour un connecteur synthétique, on aurait du faire droite/droite ou gauche/gauche.

Le lecteur non convaincu par la recherche d'une preuve "sans axiome" se convaincra en essayant de prouver le séquent suivant qui ne peut être prouvé en utilisant un axiome comme première règle :

$$\text{avec } \begin{cases} \vdash P'^\perp \otimes (Q'^\perp \wp R'^\perp), P \wp (Q \otimes R) \\ P = X_1 \otimes X_2 & Q = Y_1 \otimes Y_2 & R = Z_1 \otimes Z_2 \\ P' = X_2 \otimes X_1 & Q' = Y_2 \otimes Y_1 & R' = Z_2 \otimes Z_1 \end{cases}$$

1.1.3 Jeux

La ludique est une sémantique de jeux [AJM00, HO00] avec des propriétés particulières mettant l'accent sur la clôture par bi-orthogonalité pour définir les *types*.

Les rapports jeux/ludique sont officiels depuis longtemps [Gir99, Gir98] mais le vocabulaire de la ludique peut troubler. Un dictionnaire a été établi par Claudia Faggian et Martin Hyland [FH02]. Un consensus général décrit les desseins comme des stratégies à la Hyland et Ong [HO00] et plus précisément comme des arbres de Böhm abstraits linéaires [Cur98].

Sémantique de jeux	Ludique
type	comportement
partie	dispute
stratégie	dessein
vue	chronique
coup	action
coup joueur	action positive
coup opposant	action négative
branchement	ramification
occurrence	lieu

FIG. 1.3 – Dictionnaire sémantique de jeux/ludique.

1.1.4 Un dessein pour commencer

Le vocabulaire de la ludique est très riche et les définitions de base, bien que familières (les notions de jeux, de vues ou d'orthogonalité ont une histoire importante), ne prennent tout leur sens qu'au moment de prouver les théorèmes analytiques section 1.3 page 35.

L'option retenue a été de présenter directement un dessein figure 1.4 page suivante et de voir ensuite les définitions car un bon dessein vaut mieux qu'un long discours.

Une preuve

$$\frac{\frac{\frac{\overline{\vdash P_1, P_1^\perp} \quad \overline{\vdash P_2, P_2^\perp}}{\vdash P_1, P_2, \downarrow P_1^\perp \otimes \downarrow P_2^\perp} \quad \overline{\vdash P_3, P_3^\perp}}{\vdash P_1, P_2, P_3, \downarrow P_1^\perp \otimes \downarrow P_2^\perp \otimes \downarrow P_3^\perp}}{\vdash \uparrow P_1 \wp \uparrow P_2 \wp \uparrow P_3 \wp \uparrow (\downarrow P_1^\perp \otimes \downarrow P_2^\perp \otimes \downarrow P_3^\perp)}}$$

dont la structure de lieux (c'est ce que l'on appelle un dessin) est la suivante (les négatifs sont à gauche)

$$\frac{\frac{\frac{\vdots}{\xi.0.1 \vdash \xi.1} \quad \frac{\vdots}{\xi.0.2 \vdash \xi.2} \quad \frac{\vdots}{\xi.0.3 \vdash \xi.3}}{\vdash \xi.0, \xi.1, \xi.2, \xi.3}}{\xi \vdash}}$$

où

- Le lieu ξ correspond à la formule négative $\uparrow P_1 \wp \uparrow P_2 \wp \uparrow P_3 \wp \uparrow (\downarrow P_1^\perp \otimes \downarrow P_2^\perp \otimes \downarrow P_3^\perp)$.
- Ses sous-lieux immédiats $\xi.0, \xi.1, \xi.2$ et $\xi.3$ correspondent à ses sous-formules positives immédiates : $\downarrow P_1^\perp \otimes \downarrow P_2^\perp \otimes \downarrow P_3^\perp$, P_1 , P_2 et P_3 .
- Les sous-lieux de $\xi.0$: $\xi.0.1$, $\xi.0.2$ et $\xi.0.3$ correspondent aux formules négatives P_1^\perp , P_2^\perp et P_3^\perp et sont donc notés à gauche.

donne le dessin

$$\frac{\frac{\mathfrak{F}ar_{\xi.0.1 \vdash \xi.1} \quad \mathfrak{F}ar_{\xi.0.2 \vdash \xi.2} \quad \mathfrak{F}ar_{\xi.0.3 \vdash \xi.3}}{(+, \xi.0, \{1, 2, 3\})}}{(-, \xi, \{0, 1, 2, 3\})}}$$

que l'on note aussi

$$\begin{array}{c} \mathfrak{F}ar_{\xi.0.1 \vdash \xi.1} \quad \mathfrak{F}ar_{\xi.0.2 \vdash \xi.2} \quad \mathfrak{F}ar_{\xi.0.3 \vdash \xi.3} \\ \swarrow \quad \downarrow \quad \searrow \\ (+, \xi.0, \{1, 2, 3\}) \\ \downarrow \\ (-, \xi, \{0, 1, 2, 3\}) \end{array}$$

où

- Les dessins $\mathfrak{F}ar_{\xi.0.i \vdash \xi.i}$ correspondent aux axiomes $\overline{\vdash \uparrow P_i, \downarrow P_i^\perp}$.
- L'action $(-, \xi, \{0, 1, 2, 3\})$ est une règle négative qui correspond à l'étape des pars et des décalages. L'ensemble $\{0, 1, 2, 3\}$ qu'on appelle **ramification** correspond au branchement de la règle. 0 sera le bloc de tenseurs, 1 la formule P_1 ...
- L'action $(+, \xi.0, \{1, 2, 3\})$ est une règle positive qui correspond aux deux tenseurs et aux décalages. La ramification $\{1, 2, 3\}$ correspond aux trois formules du double tenseur.

FIG. 1.4 – Un dessin pour commencer

1.2 Lieux, actions et desseins

1.2.1 Lieux

En une étape d'interaction, la connaissance d'une formule ne peut être que partielle. Ainsi, le point de vue interactif sur les formules amène à oublier la valeur des formules pour ne s'intéresser qu'à leur structure : la structure de sous-formule. Les formules de la logique linéaire sont simplement remplacées par leurs adresses : des lieux. Cela conduit à remplacer les séquents sur les formules par des séquents sur les lieux (on appelle ces objets des desseins). Cette étape n'est pas suffisante car il reste des problèmes de séparation et donc de complétude. Elle n'est évoquée qu'à la figure 1.4 page ci-contre puis oubliée par la suite.

Définition 1.4 – Lieu

| Une **lieu** est une suite finie sur \mathbb{N} .

On note $\xi.i$ le lieu composé des entiers de ξ concaténés avec l'entier i .

Définition 1.5 – Lieux disjoints

| Deux lieux sont **disjoints** si aucun d'eux n'est le préfixe de l'autre.

Définition 1.6 – Base

| Une **base** est un ensemble de lieux disjoints deux à deux munis d'une polarité (positif/négatif) avec au plus un lieu négatif.

| Les bases sont notées β (base quelconque), $\xi \vdash \Lambda$ (base négative où ξ est l'unique lieu négatif), $\vdash \Lambda$ (base positive), ou encore $\xi_1^{\epsilon_1}, \dots, \xi_n^{\epsilon_n}$ (base quelconque avec les polarités).

| Les bases $\vdash \xi$ et $\xi \vdash$ sont dites **unaires**. La base \vdash est dite **vide**.

1.2.2 Desseins

De même que les séquents ont été *localisés* en remplaçant les formules par des lieux, les preuves sont vues comme des arbres sur de nouvelles règles : les actions. Ces arbres sont des ensembles de *chroniques*, la dénomination ludique des branches.

Les arbres ont un branchement pris en compte par la notion suivante :

Définition 1.7 – Ramification

| Une **ramification** est un ensemble fini d'entiers.

Définition 1.8 – Action

| Une **action** est

- soit positive : de la forme $(+, \xi, I)$ ou le démon noté \bowtie ,
- soit négative : de la forme $(-, \xi, I)$

où $I \subset_{fn} \mathbb{N}$ est une ramification et ξ un lieu.

Remarque 1.1 Les actions $(+, \xi, I)$ et $(-, \xi, I)$ sont duales et représentent des formules focalisées. Le démon \boxtimes n'a pas de pendant en logique linéaire mais permet de compléter l'espace des desseins. Dans la continuation de l'action (\pm, ξ, I) , les sous-lieux de ξ sont les lieux $\xi.i$ où i appartient à I .

Définition 1.9 – Justification

Une suite alternée d'actions est **justifiée** si pour tout préfixe \mathbf{c} de la suite se terminant par une action K alors

- ou bien K est le démon
- ou bien l'action K s'écrit (ϵ, ξ, I) et ξ^ϵ est dans la base β .
- ou bien l'action K s'écrit $(\epsilon, \xi.i, I)$ et il y a une action $(-\epsilon, \xi, J)$ dans \mathbf{c} avec $i \in J$.

Définition 1.10 – Pré-chronique simple

Une **pré-chronique simple** sur une base β est une suite d'actions simples de polarités alternées, justifiée, sans démon strictement à l'intérieur et telle que chaque lieu apparaît au plus une fois dans la suite. Une pré-chronique est **positive** (resp. **négative**) si elle se termine par une action positive (resp. négative).

Définition 1.11 – Chronique

Une **chronique simple** est une pré-chronique simple positive. On note Chron_β l'ensemble des chroniques sur une base β .

Des exemples de chroniques sont donnés aux figures 1.5 et 1.6 page suivante.

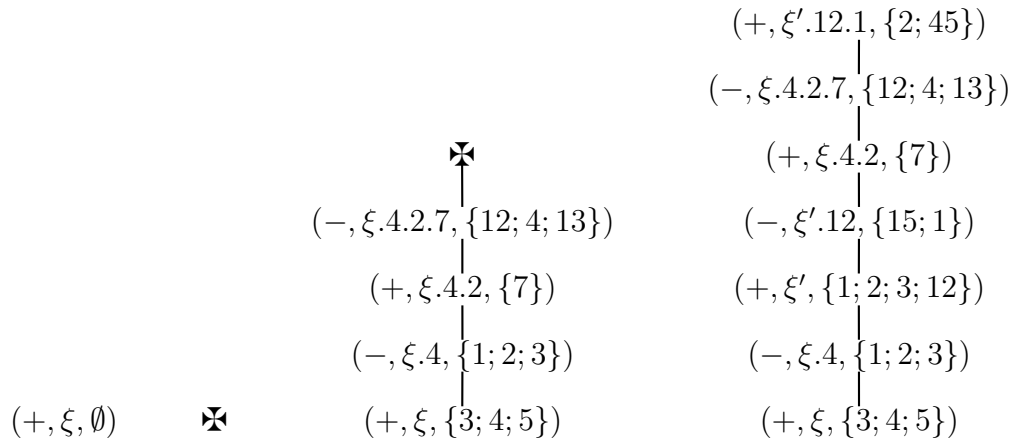


FIG. 1.5 – Exemples de chroniques positives

Définition 1.12 – Répertoire négatif

Un **répertoire négatif** est un ensemble d'actions négatives de ramifications distinctes sur un même lieu ξ .

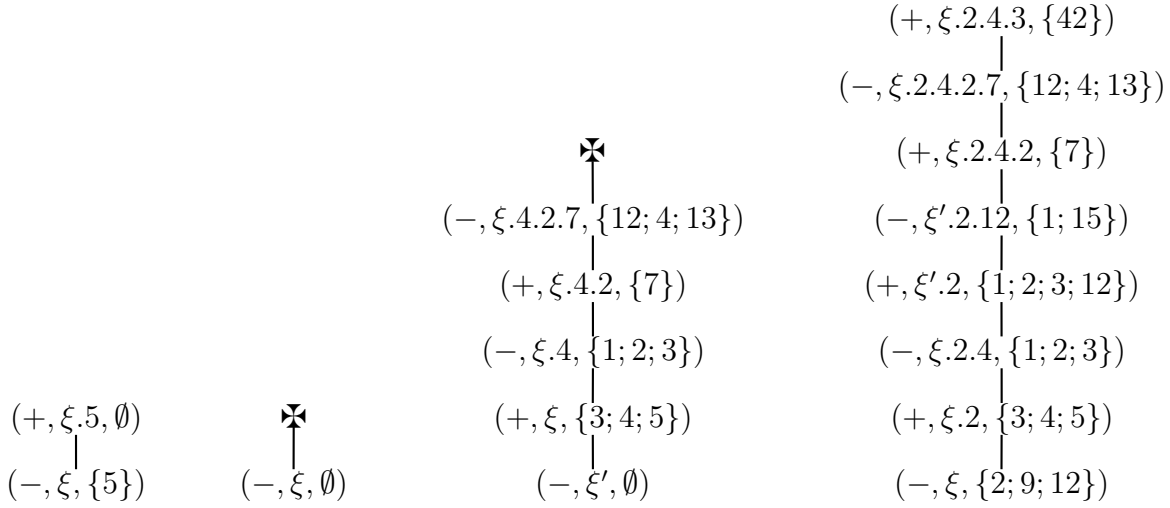


FIG. 1.6 – Exemples de chroniques négatives

Définition 1.13 – Arbre simple

Un **arbre simple** \mathcal{D} est un arbre fini ou infini engendré par la grammaire suivante :

$$\mathcal{D} := \mathcal{D}^+ \mid \mathcal{D}_\xi^-$$

$$\mathcal{D}^+ := \text{fid} \mid \mathcal{D}_t^+$$

$$\mathcal{D}_t^+ := \text{X} \mid (+, \xi, \{i_1, \dots, i_n\}).(\mathcal{D}_{\xi.i_1}^-, \dots, \mathcal{D}_{\xi.i_n}^-)$$

$$\mathcal{D}_\xi^- := ((-, \xi, I). \mathcal{D}_t^+)_{I \in \mathbb{N}}$$

où \mathbb{N} est un ensemble de sous-parties finies de \mathbb{N} .

Un arbre simple est **positif** s'il est de la forme \mathcal{D}^+ et **négatif**, s'il est de la forme \mathcal{D}_ξ^- .

Définition 1.14 – Tranche

Une **tranche** est un arbre simple dans lequel l'ensemble \mathbb{N} de chaque \mathcal{D}_ξ^- est réduit à au plus un élément.

Définition 1.15 – Ordre d'inclusion

L'**inclusion** entre arbres simples est l'ordre engendré par :

– Cas positif.

$$\mathfrak{Fid} \subseteq \mathfrak{D}$$

– Cas négatif. si $N \subseteq M$ alors

$$((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in N} \subseteq ((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in M}$$

Définition 1.16 – Tranche d'un arbre simple

Une **tranche d'un arbre simple** est une tranche incluse dans l'arbre simple.

Définition 1.17 – Linéarité des tranches

Une tranche est **linéaire** si chaque lieu d'une action de la tranche apparaît au plus une fois dans la tranche.

Définition 1.18 – Dessein

Un arbre simple **vide** est un arbre simple qui ne contient pas d'actions.

Le dessein partiel \mathfrak{Fid} est l'arbre vide sur une base positive.

Le scons \mathfrak{Sunk} est l'arbre vide sur une base négative.

Un **dessein total** sur une base β est un arbre simple non vide dont les branches sont des chroniques sur β et dont les tranches sont linéaires.

Quelques exemples de desseins sont donnés figures 1.7 et 1.8 page ci-contre. Par la suite, une syntaxe utilisée sera aussi

– $(+, \xi, I) \cdot \mathfrak{R}$, $(+, \xi, I) \cdot \mathfrak{R}_1 \mathfrak{R}_2$ ou $(+, \xi, I) \cdot (\mathfrak{D}_1, \dots, \mathfrak{D}_n)$ pour dénoter un dessein positif composé d'une action $(+, \xi, I)$ suivie d'un ensemble de desseins

$$\mathfrak{R} = \mathfrak{R}_1 \mathfrak{R}_2 = \{\mathfrak{D}_1, \dots, \mathfrak{D}_n\}.$$

– $((-, \xi, I) \cdot \mathfrak{D}_I)_{I \in \text{fin} \mathbb{N}}$ pour dénoter un dessein négatif.

1.2.3 Réseaux

Les desseins interagissent entre eux quand ils sont placés sur des lieux identiques de polarité opposée. Cette notion est celle de réseau.

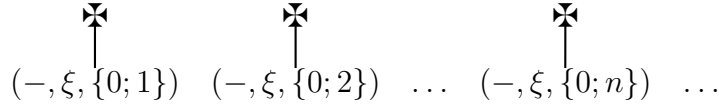
Définition 1.19 – Pré-réseau

Un **pré-réseau** est un ensemble fini de desseins sur des bases β_i tel qu'un lieu apparaît au plus dans deux bases : une fois positivement et une fois négativement.

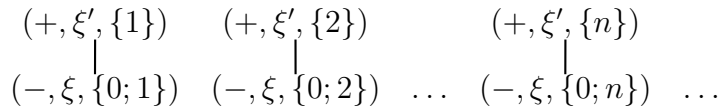
Définition 1.20 – Coupure

Une **coupure** dans un pré-réseau \mathfrak{R} est la donnée de deux desseins de \mathfrak{R} dont les bases ont un lieu en commun.

1. Un dessein négatif testant si une action jouée est de la forme $(+, \xi, \{0; n\})$ et renvoyant le démon dans ce cas. Ce dessein est infini en largeur.



2. Un dessein négatif attendant qu'une action de la forme $(+, \xi, \{0; n\})$ soit jouée et renvoyant l'action $(+, \xi', \{n\})$ dans ce cas.



3. Un dessein fondamental : le fax $\mathfrak{Fax}_{\xi \vdash \xi'}$. Le fax est une η -expansion infinie de l'identité.

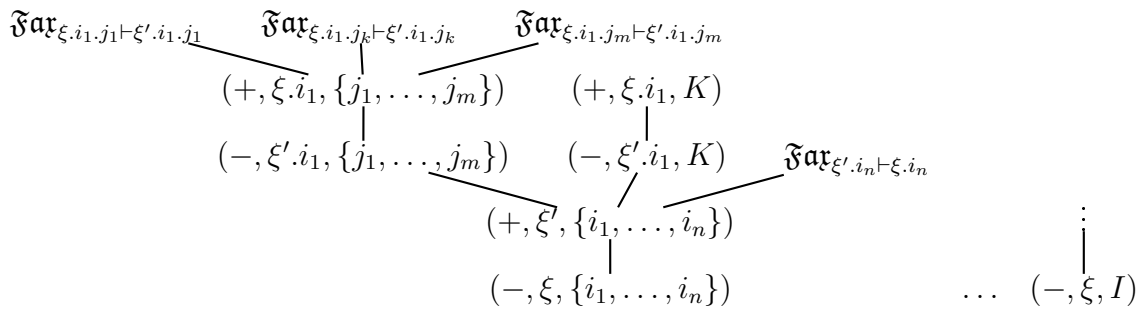


FIG. 1.7 – Exemples de desseins négatifs

1. Le dessein partiel \mathfrak{Fid} est vide.
2. Le dessein positif \mathfrak{Dai} est uniquement constitué du démon \blacktimes .
3. Le dessein positif suivant focalise sur un lieu ξ et réagit à trois réponses.

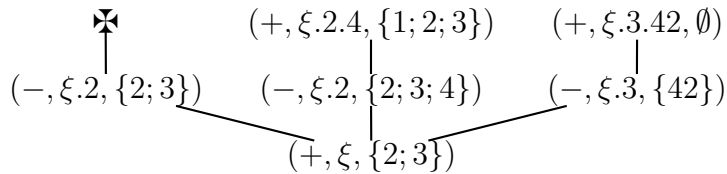


FIG. 1.8 – Exemples de desseins positifs

Définition 1.21 – Réseau

Un **réseau** est un pré-réseau dans lequel la relation de coupure est connexe et acyclique.

Définition 1.22 – Base de réseau

La **base d'un réseau** \mathfrak{R} est la base formée de l'ensemble des lieux (avec leur polarité) des bases des desseins de \mathfrak{R} qui n'apparaissent qu'une fois.

Définition 1.23 – Réseau clos

Un réseau est **clos** lorsque sa base est vide.

Définition 1.24 – Actions et lieux visibles

Soit \mathfrak{R} un réseau.

Un lieu est **visible** dans \mathfrak{R} si c'est un sous-lieu d'un lieu de la base de \mathfrak{R} .

Une action dans un dessin de \mathfrak{R} est **visible** si c'est le démon ou une action propre dont le lieu visible.

Définition 1.25 – Dessin principal

Le **dessin principal** d'un réseau est

- l'unique dessin positif de ce réseau s'il existe,
- ou l'unique dessin négatif de base $\xi \vdash \Lambda$ telle que ξ est visible.

1.2.4 Normalisation

La normalisation associe un dessin à un réseau ou diverge. Si la base est vide (cas d'un réseau clos) la normalisation du réseau renvoie le démon ou diverge car le démon est le seul dessin total sur la base vide. Il y a plusieurs méthodes équivalentes pour normaliser [Fag02] qui font intervenir des machines abstraites ou des quotients d'ordres.

Une notion fondamentale pour effectuer la normalisation est celle de tranche : elle correspond à une partie d'un dessin sans additifs [LTdF01, Gir96c].

Définition 1.26 – Normalisation d'un réseau clos

La **réduction pas à pas** d'un réseau clos \mathfrak{R} est définie par :

$$\begin{array}{lcl} K^+(\mathfrak{D}_1^- \dots \mathfrak{D}_n^-), K^-\mathfrak{D}^+, \mathfrak{R} & \rightsquigarrow & \mathfrak{D}^+, \mathfrak{D}_1^-, \dots, \mathfrak{D}_n^-, \mathfrak{R} \\ \boxtimes, \mathfrak{R} & \rightsquigarrow & \boxtimes \\ \mathfrak{R} & \rightsquigarrow & \mathfrak{Fid} \\ & & \text{dans les autres cas} \end{array}$$

La **normalisation** $\llbracket \mathfrak{R} \rrbracket$ d'un réseau clos \mathfrak{R} est définie par

$$\llbracket \mathfrak{R} \rrbracket = \begin{cases} \boxtimes & \text{si } \mathfrak{R} \rightsquigarrow^* \boxtimes \\ \mathfrak{Fid} & \text{sinon} \end{cases}$$

Si le réseau \mathfrak{R} se réduit vers \boxtimes , on dit que le réseau \mathfrak{R} **lance** le démon.

Définition 1.27 – Orthogonalité

Deux desseins \mathfrak{D} et \mathfrak{E} sont **orthogonaux** si $\llbracket \mathfrak{D}, \mathfrak{E} \rrbracket = \boxtimes$.

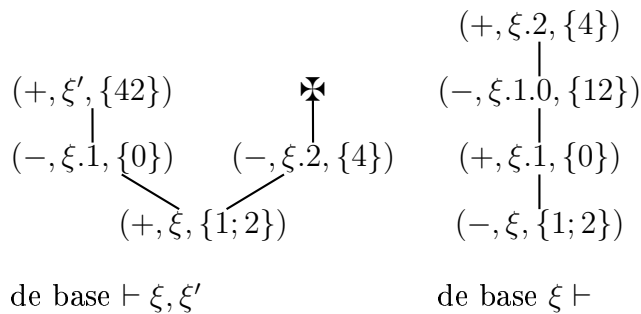
On note cette situation $\mathfrak{D} \perp \mathfrak{E}$.

On peut aussi définir la normalisation par un quotient d'ordre. Une tranche définit un ordre sur les actions. La normalisation de deux tranches agit comme un quotient de ces ordres :

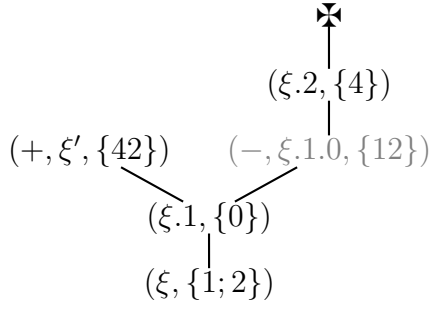
Définition 1.28 – Normalisation de tranches

La **forme normale** d'un réseau de tranches est obtenue en identifiant les actions duales, en faisant la clôture inductive des ordres des deux tranches et en renvoyant le dessin obtenu en restreignant l'ordre engendré sur les actions visibles.

Par exemple, considérons le réseau de tranches \mathfrak{R} suivant :



La fusion d'ordres de \mathfrak{R} donne le dessin de base $\vdash \xi'$:



où l'action cachée $(-, \xi.1.0, \{12\})$ est notée en grisé car elle n'a pas d'action duale et les paires d'actions duales sont notées sans polarité (ξ, I) . La normalisation consiste à enlever les paires d'actions duales ainsi que les actions grisées et leurs continuations (ce qui est plus haut dans l'arbre). Cela donne dans notre cas, le dessin réduit à l'action $(+, \xi', \{42\})$.

Définition 1.29 – Union de desseins

L'union de desseins (\mathcal{D}_k) est le dessin formé de l'union des chroniques des \mathcal{D}_k si il existe.

Définition 1.30 – Normalisation

La normalisation d'un réseau (éventuellement ouvert) est l'union des normalisations de ses tranches

$$[\mathfrak{A}] = \bigcup_{\mathfrak{G} \subseteq \mathfrak{A}} [\mathfrak{G}]$$

Dans cette définition, l'union des tranches est bien un dessin. Les différentes définitions de la normalisation donnent naturellement le même résultat.

1.2.5 Ordres

Il y a deux structures d'ordre naturelles [Cur04b] sur les desseins que l'on appelle ordre extensionnel \preceq et ordre stable \sqsubseteq .

L'ordre stable correspond à l'inclusion d'arbre tandis que l'ordre extensionnel est fondamentalement relié à la normalisation : remplacer un dessin positif par le démon le fait normaliser dans plus de réseaux.

Définition 1.31 – Ordre stable

L'ordre stable est l'ordre engendré par :

- Cas positif.

$$\mathfrak{F}id \sqsubseteq \mathcal{D}$$

- Cas négatif. si $N \subseteq M$ alors

$$((- , \xi, I) . \mathcal{D}_I)_{I \in N} \sqsubseteq ((- , \xi, I) . \mathcal{D}_I)_{I \in M}$$

Définition 1.32 – Ordre extensionnel

l'ordre extensionnel est l'ordre engendré par :

– Cas positif.

$$\mathfrak{Fid} \preceq \mathfrak{D}^+ \quad \mathfrak{D}^+ \preceq \mathfrak{X}$$

– Cas négatif. si $N \subseteq M$ alors

$$((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in N} \preceq ((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in M}$$

On résume parfois [Gir01] cet ordre par $\mathfrak{Fid} \preceq \mathfrak{D}^+ \preceq \mathfrak{X}$ en utilisant la convention d'écriture suivante : $(- , \xi, I) \cdot \mathfrak{Fid}$ signifie que l'action $(- , \xi, I)$ n'est pas présente dans le répertoire négatif considéré. Cet ordre est appelé ordre extensionnel à cause des théorèmes 1.1 et 1.5 page suivante.

1.3 Théorèmes analytiques

Les théorèmes analytiques forment une famille de quatre théorèmes qui distinguent la ludique des sémantiques de jeux habituelles. Les définitions données jusqu'ici définissent une sémantique de jeux pour l'instant assez pauvre d'un point de vue expressif : les desseins sont des stratégies linéaires. Mais elles prennent tout leur sens dans la suite avec la définition des comportements (les types) qui apparaissent grâce aux propriétés de l'orthogonalité et plus généralement de la normalisation résumées dans ces théorèmes. Ainsi, l'idée directrice de la suite de la construction est que ces théorèmes suffisent plus ou moins à aller jusqu'au théorème de complétude énonçant que les desseins "représentent" des preuves.

Définition 1.33 – Orthogonal d'un dessein

L'orthogonal d'un dessein \mathfrak{D} est

$$\mathfrak{D}^\perp = \{\mathfrak{E} \mid \mathfrak{E} \perp \mathfrak{D}\}$$

On énonce d'abord, par souci de simplicité, le théorème de séparation pour des desseins sur une base unaire.

Théorème 1.1 – Séparation

Pour tous desseins \mathfrak{D}_1 et \mathfrak{D}_2 sur une même base unaire on a :

$$\mathfrak{D}_1 = \mathfrak{D}_2 \Leftrightarrow \forall \mathfrak{E}, \llbracket \mathfrak{D}_1, \mathfrak{E} \rrbracket = \llbracket \mathfrak{D}_2, \mathfrak{E} \rrbracket$$

De plus,

$$\mathfrak{D}_1 \preceq \mathfrak{D}_2 \Leftrightarrow \mathfrak{D}_1^\perp \subseteq \mathfrak{D}_2^\perp$$

Proposition 1.2 – Séparation sur une base quelconque

Soient \mathcal{D}_1 et \mathcal{D}_2 deux desseins sur une même base. Soit χ l'ensemble des ensembles de desseins \mathfrak{R} tels que $\mathcal{D}_i, \mathfrak{R}$ soit un réseau. On a

$$\mathcal{D}_1 = \mathcal{D}_2 \Leftrightarrow \forall \mathfrak{R} \in \chi, \llbracket \mathcal{D}_1, \mathfrak{R} \rrbracket = \llbracket \mathcal{D}_2, \mathfrak{R} \rrbracket$$

Le théorème de séparation se prouve en exhibant des testeurs de chroniques et en les appliquant sur les chroniques de longueur minimale parmi celles qui n'apparaissent pas dans l'autre dessin ; le point fondamental étant qu'un dessin est isomorphe à l'ensemble de ses chroniques.

Définition 1.34 – Intersection

L'**intersection** d'une famille (\mathcal{D}_i) de desseins $\bigcap_i \mathcal{D}_i$ est le dessin formé des chroniques incluses dans chaque \mathcal{D}_i .

On étend naturellement cette notion aux réseaux.

Théorème 1.3 – Stabilité

Soit (\mathfrak{R}_i) une famille de réseaux inclus dans un réseau \mathfrak{R} . On a

$$\llbracket \bigcap \mathfrak{R}_i \rrbracket = \bigcap \llbracket \mathfrak{R}_i \rrbracket$$

Pour pouvoir parler de notion de confluence (ici d'associativité), il faut partitionner les réseaux :

Définition 1.35 – Réseau de réseaux

Un **réseau de réseaux** $\{\mathfrak{R}_1, \dots, \mathfrak{R}_n\}$ est une partition d'un réseau en réseaux.

Dans un réseau de réseaux, chaque \mathfrak{R}_i est un réseau et l'union des \mathfrak{R}_i est elle-même un réseau.

Théorème 1.4 – Associativité

Pour un réseau de réseaux $\{\mathfrak{R}_1, \dots, \mathfrak{R}_n\}$, on a

$$\llbracket \mathfrak{R}_1, \dots, \mathfrak{R}_n \rrbracket = \llbracket \llbracket \mathfrak{R}_1 \rrbracket, \dots, \llbracket \mathfrak{R}_n \rrbracket \rrbracket$$

Théorème 1.5 – Monotonie

La normalisation est croissante vis-à-vis de l'ordre extensionnel \preceq .

$$\forall i, \mathcal{D}_i \preceq \mathcal{D}'_i \implies \llbracket \mathcal{D}_1, \dots, \mathcal{D}_n \rrbracket \preceq \llbracket \mathcal{D}'_1, \dots, \mathcal{D}'_n \rrbracket$$

1.4 Comportements

On se place sur une base unaire car cela simplifie la notion d'orthogonal. Les comportements correspondent aux types. Le point fondamental est que leur définition est faite par orthogonalité.

Définition 1.36 – Éthique

Une **éthique** est un ensemble E de desseins totaux sur une même base.

Définition 1.37 – Orthogonal d'une éthique

L'**orthogonal** d'une éthique est

$$E^\perp = \{\mathfrak{D} \mid \forall \mathfrak{E} \in E, \mathfrak{D} \perp \mathfrak{E}\}$$

Définition 1.38 – Comportement

Un **comportement** est une éthique égale à son bi-orthogonal.

Remarque 1.2 L'orthogonal d'une éthique est un comportement.

On remarque (par définition de \preceq) qu'un comportement est clos par l'ordre extensionnel et donc par l'ordre stable :

Proposition 1.6 – Clôture

Soit G un comportement.

- Si $\mathfrak{D} \preceq \mathfrak{D}'$ et $\mathfrak{D} \in G$ alors $\mathfrak{D}' \in G$.
- En particulier, si $\mathfrak{D} \sqsubseteq \mathfrak{D}'$ et $\mathfrak{D} \in G$ alors $\mathfrak{D}' \in G$.
- Soit (\mathfrak{D}_k) une famille de desseins dans un comportement G , si $\bigcup_k \mathfrak{D}_k$ est un dessein alors $\bigcap_k \mathfrak{D}_k$ est un dessein et appartient à G .

Il est alors naturel de considérer les générateurs d'un comportement. La notion importante de ce point de vue est celle d'incarnation (élément minimal pour l'ordre stable).

Proposition 1.7 – Incarnation

Pour tout comportement G , et tout dessein \mathfrak{D} de G , il existe un unique dessein $\mathfrak{D}_0 \subseteq \mathfrak{D}$ de G tel que

$$\forall \mathfrak{E} \in G, \mathfrak{E} \subseteq \mathfrak{D} \Rightarrow \mathfrak{E} = \mathfrak{D}_0$$

De plus, ce dessein \mathfrak{D}_0 est l'intersection des desseins de G inclus dans \mathfrak{D} .

Définition 1.39 – Incarnation

L'**incarnation** d'un dessein \mathfrak{D} dans un comportement G est le dessein de la proposition 1.7.

$$|\mathfrak{D}|_G = \bigcap \{\mathfrak{E} \in G \mid \mathfrak{E} \subseteq \mathfrak{D}\}$$

Remarque 1.3 *Naturellement, l'incarnation vérifie la propriété suivante :*

$$\forall \mathfrak{E} \in \mathcal{G}, \mathfrak{E} \subseteq \mathfrak{D} \Rightarrow |\mathfrak{D}|_{\mathcal{G}} \subseteq \mathfrak{E}$$

Définition 1.40 – Dessen incarné

Un dessein est **incarné** dans un comportement si il est égal à son incarnation dans ce comportement. On dit aussi qu'un dessein incarné est **matériel**.

Définition 1.41 – Incarnation d'un comportement

L'**incarnation d'un comportement**, notée $|G|$, est l'ensemble de ses desseins incarnés.

Les connecteurs devraient être des fonctions des comportements dans les comportements mais les définitions simples de connecteurs ne sont pas toujours closes par bi-orthogonal. Cependant, les connecteurs ne sont pas quelconques. Certains conservent une propriété de clôture plus faible : celle d'éthique complète.

Définition 1.42 – Éthique complète

Une éthique est **complète** si $|E^{\perp\perp}| \subseteq E$.

Proposition 1.8 – Éthique complète

- Une éthique complète est un comportement si et seulement si elle est close vers le haut suivant l'ordre stable.
- La clôture par bi-orthogonal d'une éthique complète est sa clôture vers le haut par l'ordre stable.

Démonstration : Le premier point découle du second. Celui-ci est prouvé en remarquant que l'incarnation d'un comportement est l'ensemble de ses éléments minimaux pour l'ordre stable. ■

Un comportement est une éthique complète. Les connecteurs associent naturellement une éthique complète à une paire de comportements mais pas forcément directement un comportement. Ainsi, les notions d'incarnation et d'éthique complète prennent tout leur intérêt dans la définition 1.50 page 41 du connecteur $G \& H$ qui est un produit cartésien (dans certaines conditions) des incarnations de G et H : la complétude est atteinte pour les incarnations.

1.5 Connecteurs

Les connecteurs de la ludique sont *locatifs* : ils sont définis en prenant en compte la structure des lieux. Cela amène naturellement à des connecteurs partiels : ils ne sont définis qu'entre comportements disjoints (définition 1.47 page 40). Les connecteurs habituels de la logique linéaire sont retrouvés grâce à des *délocalisations*.

Définition 1.43 – Délocalisation

Une **délocalisation** de type $\xi \rightarrow \xi'$ est une paire $(\delta, (\delta_\sigma)_\sigma)$ où δ est une fonction injective des sous-lieux de ξ dans les sous-lieux de ξ' et, pour tout sous-lieu σ de ξ , δ_σ est une fonction des entiers dans les entiers telles que

- $\delta(\xi) = \delta(\xi')$
- $\delta(\sigma.i) = \delta(\sigma).\delta_\sigma(i)$

Une délocalisation se relève naturellement aux desseins et aux comportements. Un exemple typique de délocalisation est fourni par le $\mathfrak{F}\alpha_{\xi \rightarrow \xi'}$ (figure 1.7 page 31) qui implémente une délocalisation de ξ dans ξ' . Ainsi, pour tout dessein \mathfrak{D} sur la base $\vdash \xi$, le réseau \mathfrak{D} , $\mathfrak{F}\alpha_{\xi \rightarrow \xi'}$ normalise vers $\delta(\mathfrak{D})$ où δ est la délocalisation de ξ dans ξ' telle que pour tout σ , δ_σ est l'identité.

1.5.1 Décalages

Les décalages sont des connecteurs très simples de la ludique : ils permettent de changer de polarité en rajoutant une action de polarité convenable avant chaque dessein.

Définition 1.44 – Décalage

L'opération de **décalage** est définie pour les chroniques, les desseins et les comportements.

- Si le lieu ξ est disjoint de lieux Λ , le **décalage** (sur le lieu $\xi.i$) d'une chronique \mathfrak{c} sur une base $\vdash \xi.i, \Lambda$ (resp. $\xi.i \vdash \Lambda$) est la chronique $(-, \xi, \{i\}).\mathfrak{c}$ (resp. $(+, \xi, \{i\}).\mathfrak{c}$) sur la base $\xi \vdash \Lambda$ (resp. $\vdash \xi, \Lambda$).
- Cette construction s'étend immédiatement aux desseins comme ensembles de chroniques avec de plus

$$\uparrow \mathfrak{F}id = \mathfrak{G}\mathfrak{k}\mathfrak{u}\mathfrak{n}\mathfrak{k} \text{ et } \downarrow \mathfrak{G}\mathfrak{k}\mathfrak{u}\mathfrak{n}\mathfrak{k} = (+, \xi, \{i\})$$

- Sur une base unaire, le **décalage** d'un comportement est le bi-orthogonal des décalages de ses desseins. Lorsque G est positif de base $\vdash \xi.i$ (resp. négatif de base $\xi.i \vdash$), on le note $\uparrow G$ (resp. $\downarrow G$).

Proposition 1.9 – Complétude interne pour les décalages

Soit G un comportement.

- Si G est positif, alors $\{\uparrow \mathfrak{D} \mid \mathfrak{D} \in G\}$ est une éthique complète pour $\uparrow G$.
- Si G est négatif, alors $\downarrow G = \{\downarrow \mathfrak{D} \mid \mathfrak{D} \in G\} \cup \{\mathfrak{X}\}$.

1.5.2 Additifs

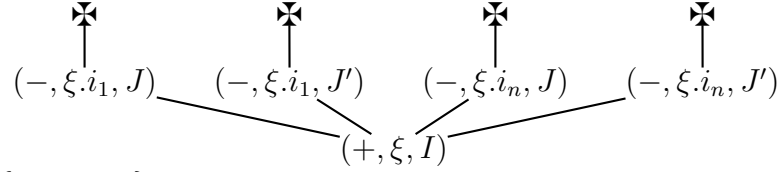
Le connecteur *avec* est une intersection et le connecteur *plus* une union. Ils peuvent être définis sur des cardinaux arbitraires mais par simplicité, la définition donnée ici est restreinte au cas binaire.

Définition 1.45 – Répertoire

Un **répertoire** est un ensemble de ramifications.

Définition 1.46 – Répertoire d'un comportement

- Le **répertoire** d'un comportement positif G est l'ensemble des ramifications I telles que le dessein suivant appartienne à G



où $\begin{cases} I = \{i_1, \dots, i_n\} \\ J, J' \text{ parcourent } \mathcal{P}_{fin}(\mathbb{N}) \end{cases}$

- Le **répertoire** d'un comportement négatif est l'ensemble des ramifications I

telles que le dessein $\begin{matrix} \uparrow \otimes \\ (-, \xi, I) \end{matrix}$ est inclus dans l'incarnation dans G de

$\begin{matrix} \uparrow \otimes \\ (-, \xi, J) \end{matrix} \cdots \begin{matrix} \uparrow \otimes \\ (-, \xi, J') \end{matrix}$ où J, J' parcourent $\mathcal{P}_{fin}(\mathbb{N})$.

- le **répertoire** d'une éthique est le répertoire de son bi-orthogonal.

Proposition 1.10 – Répertoire

Tout comportement sur une base unaire a le même répertoire que son orthogonal.

Définition 1.47 – Comportements disjoints et connexes

- Deux comportements sont **disjoints** si leurs répertoires sont disjoints.
- Une éthique E est **connexe** si son répertoire est un singleton $\{I\}$. L'ensemble I est appelé la **ramification** de E .

Proposition 1.11 – Caractérisation des comportements disjoints

- Deux comportements positifs sont disjoints si et seulement si leur intersection est réduite au démon.
- Deux comportements négatifs sont disjoints si et seulement si

$$\forall \mathcal{D} \in G \forall \mathcal{E} \in H, |\mathcal{D}|_G \cap |\mathcal{E}|_H = \emptyset.$$

Définition 1.48 – Le connecteur \cap

Soit (G_k) une famille de comportements sur une même base. On définit le comportement $\bigcap_k G_k$ par : $\bigcap_k G_k = \{\mathcal{D} \mid \forall k, \mathcal{D} \in G_k\}$.

C'est bien un comportement, indépendamment de la polarité.

Définition 1.49 – Le connecteur $\bigcup^{\perp\perp}$

Soit (G_k) une famille de comportements sur une même base. On définit le comportement $\bigcup_k^{\perp\perp} G_k$ par : $\bigcup_k^{\perp\perp} G_k = \{\mathfrak{D} \mid \exists k, \mathfrak{D} \in G_k\}^{\perp\perp}$.

On définit maintenant un connecteur partiel :

Définition 1.50 – Le connecteur avec $\&$

Le connecteur $\&$ est le connecteur \cap appliqué à des comportements négatifs disjoints.

Le connecteur $\&$ est un produit cartésien sur les incarnations :

Proposition 1.12 – Mystère de l'incarnation

Pour tous comportements disjoints G et H sur une même base négative,

$$|G \& H| = |G| \times |H|$$

Le connecteur habituel (appelé **spirituel** dans [Gir01]) étant la pré-composition de ce connecteur partiel par des délocalisations disjoints (c'est-à-dire d'images disjoints).

Définition 1.51 – Le connecteur *plus* \oplus

Le connecteur \oplus est le connecteur $\bigcup^{\perp\perp}$ appliqué à des comportements positifs disjoints.

Proposition 1.13 – Propriété de la disjonction

Pour tous comportements positifs disjoints,

$$G \oplus H = G \cup H$$

Proposition 1.14 – Décomposition additive

Tout comportement positif se décompose de manière unique comme une somme de comportements connexes.

Tout comportement négatif se décompose de manière unique comme un avec de comportements connexes.

1.5.3 Multiplicatifs

Le *tenseur* est un connecteur positif. A cause de la focalisation, le *tenseur* en ludique entre deux desseins doit agréger leurs premières actions. Dans [Gir01], quatre *tenseurs* sont définis suivant la manière d'agréger ces premières actions. Ils sont indifféremment notés \otimes . Dans des conditions particulières, ces *tenseurs* sont égaux et sont ainsi notés \otimes .

La définition du *tenseur* est revue au chapitre 2 page 53 car la version présentée ici [Gir01] est problématique pour le dessin partiel \mathfrak{Fid} .

On se place dans le cadre de la base unaire $\vdash \xi$ ou $\xi \vdash$ suivant la polarité.

Définition 1.52 – Tenseur

Le tenseur de deux desseins positifs totaux \mathcal{D}_1 et \mathcal{D}_2 est défini par cas sur la première action :

- Si un des desseins est le démon, le tenseur est le démon.
- Si $\mathcal{D}_1 = (+, \xi, I_1).\mathcal{R}_1$ et $\mathcal{D}_2 = (+, \xi, I_2).\mathcal{R}_2$. Il y a deux cas :
 - Si $I_1 \cap I_2 = \emptyset$ alors $\mathcal{D}_1 \otimes \mathcal{D}_2 = (+, \xi, I_1 \cup I_2).\mathcal{R}_1\mathcal{R}_2$
 - Si $I_1 \cap I_2 \neq \emptyset$ alors il y a quatre cas suivant la définition du tenseur :
 - Tenseur commutatif minimal : $\mathcal{D}_1 \otimes \mathcal{D}_2 = \mathfrak{Fid}$;
 - Tenseur commutatif maximal : $\mathcal{D}_1 \otimes \mathcal{D}_2 = \mathfrak{K}$;
 - Tenseur non-commutatif gauche : $\mathcal{D}_1 \otimes \mathcal{D}_2 = (+, \xi, I_1 \cup I_2).\mathcal{R}_1\mathcal{R}'_2$ où \mathcal{R}'_2 est égal à \mathcal{R}_2 privé des desseins commençant par des actions sur des lieux dans $\{\xi.i \mid i \in I_1\}$;
 - Tenseur non-commutatif droite : $\mathcal{D}_1 \otimes \mathcal{D}_2 = (+, \xi, I_1 \cup I_2).\mathcal{R}_2\mathcal{R}'_1$ où \mathcal{R}'_1 est égal à \mathcal{R}_1 privé des desseins commençant par des actions sur des lieux dans $\{\xi.i \mid i \in I_2\}$.

Ces tenseurs possèdent un adjoint comme défini ici :

Définition 1.53 – Adjonctions

Pour tout tenseur \otimes et pour tous desseins $\mathcal{D}_1, \mathcal{D}_2$ et \mathcal{E} sur les bases respectives $\vdash \xi, \vdash \xi$ et $\xi \vdash$, il existe deux uniques desseins $\mathcal{E}^{\mathcal{D}_1}$ et $\mathcal{E}_{\mathcal{D}_2}$, les **adjoints**, sur la base $\xi \vdash$ tels que

$$\llbracket \mathcal{E}, \mathcal{D}_1 \otimes \mathcal{D}_2 \rrbracket = \llbracket \mathcal{E}^{\mathcal{D}_1}, \mathcal{D}_2 \rrbracket = \llbracket \mathcal{E}_{\mathcal{D}_2}, \mathcal{D}_1 \rrbracket$$

De plus, $\mathcal{E}^{\mathcal{D}_1}$ ne dépend que de \mathcal{E} et \mathcal{D}_1 . De même, $\mathcal{E}_{\mathcal{D}_2}$ ne dépend que de \mathcal{E} et \mathcal{D}_2 .

Cette propriété d'adjonction permet en particulier de montrer l'associativité des tenseurs.

Définition 1.54 – Tenseur de comportements

Le tenseur de deux comportements positifs est défini par

$$G_1 \otimes G_2 = \{\mathcal{D}_1 \otimes \mathcal{D}_2 \mid \mathcal{D}_1 \in G_1, \mathcal{D}_2 \in G_2\}^{\perp\perp}$$

Je me contente de définir un connecteur multiplicatif négatif (le nœud papillon) qui est réutilisé à la définition 1.71 page 46.

Définition 1.55 – Le connecteur \bowtie

Le nœud papillon \bowtie est le dual du tenseur commutatif maximal.

Remarque 1.4 La définition directe de ce connecteur n'est pas aisée, le plus simple est de passer par l'orthogonal.

Pour les additifs, la condition de disjonction (définition 1.47 page 40) donne la complétude. Il y a une condition similaire pour les multiplicatifs : la notion de comportements étrangers (définition 1.58 page ci-contre).

Définition 1.56 – Réservoir

Un **réservoir** est un ensemble d'entiers.

Le **réservoir** d'un comportement est l'union des ramifications de son répertoire.

Proposition 1.15 – Existence d'une projection

Soit $\mathbb{M} \subseteq \mathbb{N}$ un réservoir. Tout dessein positif $\mathcal{D} = (+, \xi, I). \mathfrak{R}$ est égal à $\mathcal{D}_{\mathbb{M}} \otimes \mathcal{D}_{\mathbb{N} \setminus \mathbb{M}}$ où $\mathcal{D}_{\mathbb{M}}$ commence par $(+, \xi, I \cap \mathbb{M})$ et $\mathcal{D}_{\mathbb{N} \setminus \mathbb{M}}$ commence par $(+, \xi, I \setminus \mathbb{M})$.

Le dessein $\mathcal{D}_{\mathbb{M}}$ est le **projeté** de \mathcal{D} sur \mathbb{M} .

On étend cette définition en posant $\mathfrak{X}_{\mathbb{M}} = \mathfrak{X}$.

Démonstration : L'idée de la construction de $\mathcal{D}_{\mathbb{M}}$ est simplement d'écrire

$$\mathcal{D} = (+, \xi, I).(\mathcal{D}_{\xi.i_1}, \dots, \mathcal{D}_{\xi.i_n})$$

et de ne conserver que les $\mathcal{D}_{\xi.i_k}$ avec $i_k \in \mathbb{M}$. ■

Une propriété de la projection qui sera fondamentale pour la suite est énoncée dans la proposition suivante :

Proposition 1.16 – Projection

Soit E une éthique connexe et \mathbb{M} un réservoir. On a $(E^{\perp\perp})_{\mathbb{M}} = (E_{\mathbb{M}})^{\perp\perp}$.

Définition 1.57 – Indépendance

Deux répertoires R_1 et R_2 sont **indépendants** si

$$\forall I_1, J_1 \in R_1, I_2, J_2 \in R_2, I_1 \cup I_2 = J_1 \cup J_2 \Rightarrow I_1 = J_1, I_2 = J_2$$

Deux comportements sont **indépendants** si leurs répertoires sont indépendants.

Proposition 1.17 – Complétude interne pour le tenseur

Si G_1 et G_2 sont positifs et indépendants alors $\{\mathcal{D}_1 \otimes \mathcal{D}_2 \mid \mathcal{D}_1 \in G_1, \mathcal{D}_2 \in G_2\}$ est une éthique complète pour $G_1 \otimes G_2$.

Définition 1.58 – Comportements étrangers

Deux comportements sont **étrangers** si leurs réservoirs sont disjoints.

Deux comportements étrangers sont indépendants. De plus, dans ce cas, les quatre tenseurs sont égaux. Le résultat est noté $G_1 \otimes G_2$. C'est le principal intérêt de cette notion de comportements étrangers. Dans ces conditions, l'adjoint est lui-même noté $\mathfrak{F}(\mathcal{D})$.

1.6 Quantificateurs

On ne s'occupe que des quantificateurs du second ordre car les quantificateurs du premier ordre ne sont pas aisément adaptables à la ludique.

Une extension au premier ordre a été apportée par Marie-Renée Fleury et Myriam Quatrini [FQ04]. Cette solution, bien qu'élégante à plusieurs aspects, manque d'*interactivité* et semble difficile à généraliser (au cadre probabiliste par exemple). C'est pourquoi les quantificateurs du premier ordre ne sont pas pris en compte dans cette thèse.

Le quantificateur universel est simplement une intersection de comportements alors que le quantificateur existentiel est une union close par bi-orthogonal. La simplicité de la définition — l'intersection — amène de nouvelles propriétés : les formes prenexes. Ainsi, le quantificateur universel commute à tous les autres connecteurs à part le quantificateur existentiel...

1.7 Uniformité

L'uniformité est la notion ludique qui correspond aux relations logiques [Plo80]. Elle permet de sélectionner les desseins d'un comportement comme gagnants ou perdants. Un dessein est uniforme s'il est équivalent à lui-même. Dans le cadre uniforme, les comportements sont remplacés par les *bi-comportements* (définition 1.66 page suivante). Naturellement, les connecteurs agissent maintenant sur les bi-comportements.

1.7.1 Bi-comportements

Définition 1.59 – Relation d'équivalence partielle

Une **relation d'équivalence partielle** (PER) est une relation symétrique et transitive.

Définition 1.60 – Dessein partiel

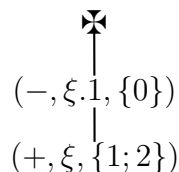
La divergence \mathfrak{Fid} est appelé le **dessein partiel**.
Un **dessein partiel d'une éthique** E est un dessein éventuellement partiel inclus dans un dessein de E .

Définition 1.61 – Éthique partielle

Soit E une éthique. L'**éthique partielle** associée E^p est l'ensemble des desseins partiels de E .

On définit de même la notion de comportement partiel.

Un exemple de comportement G est le comportement formé du démon \mathfrak{X} et de tous les desseins contenant le dessein \mathfrak{D} :



Le comportement partiel G^p est l'ensemble des desseins constitués du démon \mathfrak{X} , de la divergence \mathfrak{Fid} , des desseins contenant

$$\begin{array}{c}
 \blackcross \\
 \uparrow \\
 (-, \xi.1, \{0\}) \\
 \downarrow \\
 (+, \xi, \{1; 2\})
 \end{array}$$

ainsi que des desseins dont la première action est $(+, \xi, \{1; 2\})$ et qui ne contiennent pas l'action $(-, \xi.1, \{0\})$.

Définition 1.62 – La relation \equiv_G

Sur un comportement partiel G^p , on définit l'équivalence \equiv_G par

$$\mathcal{D}_1 \equiv_G \mathcal{D}_2 \Leftrightarrow \forall \mathcal{E} \in G^\perp, \llbracket \mathcal{D}_1, \mathcal{E} \rrbracket = \llbracket \mathcal{D}_2, \mathcal{E} \rrbracket$$

On peut alors définir une notion d'incarnation adaptée à cette équivalence. Cette notion préfigure celle de bi-incarnation (définition 1.68 page suivante).

Définition 1.63 – Incarnation dans un comportement partiel

$$|\mathcal{D}|_G = \bigcap \{ \mathcal{D}' \mid \mathcal{D}' \subseteq \mathcal{D}, \mathcal{D}' \equiv_G \mathcal{D} \}$$

Définition 1.64 – Bi-éthique

Une **bi-éthique** sur une base β est une paire (E, \equiv) formée d'une éthique E et d'une relation d'équivalence partielle \equiv sur E^p telle que

– Cas 1. La base β est positive.

Si $\mathcal{D} \equiv \mathcal{D}'$ alors :

– si $\mathcal{D} = \mathfrak{F}id$ alors $\mathcal{D}' = \mathfrak{F}id$;

– si $\mathcal{D} = \blackcross$ alors $\mathcal{D}' = \blackcross$.

– Cas 2. La base β est négative.

$\mathfrak{G}kun\bar{k} \equiv \mathfrak{G}kun\bar{k}$ et $\mathcal{D}ai^- \equiv \mathcal{D}ai^-$.

Définition 1.65 – Orthogonal d'une bi-éthique

L'**orthogonal d'une bi-éthique** (E, \equiv) est (E^\perp, \equiv^\perp) où $\mathcal{E} \equiv^\perp \mathcal{E}'$ si

$$\forall \mathcal{D}, \mathcal{D}' \in E, \mathcal{D} \equiv \mathcal{D}' \Rightarrow \llbracket \mathcal{D}, \mathcal{E} \rrbracket = \llbracket \mathcal{D}', \mathcal{E}' \rrbracket$$

L'orthogonal d'une bi-éthique est une bi-éthique.

Définition 1.66 – Bi-comportement

Un **bi-comportement** est une bi-éthique égale à son bi-orthogonal.

Définition 1.67 – Uniformité

Un dessin \mathcal{D} est **uniforme** dans un bi-comportement (G, \equiv) si $\mathcal{D} \equiv \mathcal{D}$.

Définition 1.68 – Bi-incarnation

La **bi-incarnation** d'un dessein uniforme \mathcal{D} dans un bi-comportement (G, \equiv) est

$$\|\mathcal{D}\|_G = \bigcap \{\mathcal{D}' \mid \mathcal{D}' \subseteq \mathcal{D}, \mathcal{D} \equiv \mathcal{D}'\}$$

Un dessein égal à sa bi-incarnation est dit **bi-matériel** ou **bi-incarné**.

1.7.2 Connecteurs

On définit l'action des connecteurs sur les bi-comportements.

Les additifs sont définis naturellement : l'intersection est un quantificateur universel et l'union nécessite une clôture par bi-orthogonal.

Définition 1.69 – Le connecteur \bigcap

L'équivalence partielle $\equiv_{\bigcap_k G_k}$ est définie par

$$\mathcal{D} \equiv_{\bigcap_k G_k} \mathcal{D}' \Leftrightarrow \forall k, \mathcal{D} \equiv_{G_k} \mathcal{D}'$$

Définition 1.70 – Le connecteur \bigcup

L'équivalence partielle $\equiv_{\bigcup_k G_k}$ est la clôture transitive de l'union des \equiv_{G_k} .

Le comportement $(\bigcup_k G_k, \equiv_{\bigcup_k G_k})$ est sa clôture par bi-orthogonal.

Les multiplicatifs sont définis à partir du négatif qui agit comme une fonction (cela provient de l'égalité de la logique linéaire $A \multimap B = A^\perp \wp B$).

Définition 1.71 – Le connecteur \bowtie

Soient (G, \equiv_G) et (H, \equiv_H) deux bi-comportements sur la base $\xi \vdash$. L'équivalence partielle $\equiv_{G \bowtie H}$ est définie par

$$\mathfrak{F}_1 \equiv_{G \bowtie H} \mathfrak{F}_2 \Leftrightarrow \forall \mathcal{D}_1 \mathcal{D}_2, (\mathcal{D}_1 \equiv_{G^\perp} \mathcal{D}_2 \Rightarrow \mathfrak{F}_1^{\mathcal{D}_1} \equiv_H \mathfrak{F}_2^{\mathcal{D}_2})$$

Remarque 1.5 Le connecteur \bowtie est défini simplement comme le dual d'un tenseur (remarque 1.4 page 42) alors que c'est l'équivalence partielle sur le tenseur qui est définie simplement à partir de celle sur le \bowtie .

Définition 1.72 – Dessein propre

Un dessein est **propre** s'il est différent de $\mathfrak{F}id$ et $\mathcal{D}ai$.

Proposition 1.18 – Tenseur de comportements positifs et étrangers

Soient (G, \equiv_G) et (H, \equiv_H) deux bi-comportements étrangers et positifs. Pour tous desseins propres $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}'_1$ et \mathcal{D}'_2 on a :

$$\mathcal{D}_1 \otimes \mathcal{D}_2 \equiv_{G \otimes H} \mathcal{D}'_1 \otimes \mathcal{D}'_2 \Leftrightarrow (\mathcal{D}_1 \equiv_G \mathcal{D}'_1 \text{ et } \mathcal{D}_2 \equiv_H \mathcal{D}'_2)$$

Remarque 1.6 *La proposition précédente n'est pas généralisable aux desseins \mathfrak{Fid} et \mathfrak{Dai} car ils sont absorbants pour le tenseur. Cette condition n'est pas donnée dans [Gir01]. Pour la complétude, la condition de gain (définition 1.79 page suivante) impose que les desseins soient propres.*

Cette proposition est l'archétype des propositions sur l'uniformité : le tenseur de deux desseins propres est uniforme si et seulement si chaque dessein est uniforme. Cela implique (définition 1.79 page suivante) qu'un tenseur de desseins est gagnant si et seulement si chaque dessein est gagnant.

1.8 Séquents de comportements

On a pour l'instant décrit le cadre pour interpréter les formules et leurs preuves. Il faut encore interpréter les séquents.

Définition 1.73 – Séquent de comportements

Un **séquent de comportements** sur une base $\Xi \vdash \Lambda$ notée $(\xi_i^{\epsilon_i})$ est un ensemble $\{\Xi, \Lambda\}$ de comportements G_i sur les bases $\vdash \xi_i$.
Le comportement $\Xi \vdash \Lambda$ sur la base $\Xi \vdash \Lambda$ est l'orthogonal des familles de desseins (\mathcal{D}_{ξ_i}) où $\mathcal{D}_{\xi_i} \in G_i$ si $\xi_i \in \Xi$ et $\mathcal{D}_{\xi_i} \in G_i^\perp$ sinon.

Proposition 1.19 – Exemples de séquents de comportements

Quelques séquents de comportements :

- \vdash est réduit à $\{\mathfrak{Dai}\}$
- $\vdash G$ est égal à G .
- $G \vdash$ est égal à G^\perp .
- $\mathcal{D} \in G \vdash \Lambda \Leftrightarrow \forall \mathfrak{E} \in G, \llbracket \mathcal{D}, \mathfrak{E} \rrbracket \in \vdash \Lambda$
- $\mathcal{D} \in \Xi \vdash G, \Lambda \Leftrightarrow \forall \mathfrak{E} \in G^\perp, \llbracket \mathcal{D}, \mathfrak{E} \rrbracket \in \Xi \vdash \Lambda$

L'uniformité doit encore être définie pour les séquents de comportements.

Définition 1.74 – Séquent de bi-comportements

Un **séquent de bi-comportements** sur une base $\Xi \vdash \Lambda$ notée $(\xi_i^{\epsilon_i})$ est un ensemble $\{\Xi, \Lambda\}$ de bi-comportements (G_i, \equiv_i) sur les bases $\vdash \xi_i$.
Le bi-comportement $\Xi \vdash \Lambda$ sur la base $\Xi \vdash \Lambda$ est l'orthogonal des familles de desseins (\mathcal{D}_{ξ_i}) où $\mathcal{D}_{\xi_i} \in G_i$ si $\xi_i \in \Xi$ et $\mathcal{D}_{\xi_i} \in G_i^\perp$ sinon, avec l'équivalence partielle

$$\mathcal{D} \equiv \mathcal{D}' \Leftrightarrow (\forall (\mathfrak{E}_{\xi_i}), \forall (\mathfrak{E}'_{\xi_i}), \forall i, \mathfrak{E}_{\xi_i} \equiv_i^{\epsilon_i^\perp} \mathfrak{E}'_{\xi_i} \Rightarrow \llbracket \mathcal{D}, (\mathfrak{E}_{\xi_i}) \rrbracket \equiv \llbracket \mathcal{D}', (\mathfrak{E}'_{\xi_i}) \rrbracket)$$

$$\text{où } \equiv_i^{\epsilon_i^\perp} = \begin{cases} \equiv_i & \text{si } \epsilon_i = - \\ \equiv_i^\perp & \text{si } \epsilon_i = + \end{cases}$$

1.9 Complétude

Pour l'étude de la complétude, nous commençons par définir la notion de dessein gagnant.

Définition 1.75 – Obstination

| Un dessein est **obstiné** s'il n'utilise pas le démon.

La notion d'exactitude est liée à la linéarité (par opposition au caractère affine). Intuitivement, elle stipule que tous les lieux créés dans un dessein doivent être utilisés. Elle peut être définie à l'aide de dessins [Gir01] ou de tranches :

Définition 1.76 – Tranche exacte

| Une tranche est **exacte** si elle est de la forme

- \boxtimes
- $(+, \xi, \emptyset)$
- $(+, \xi, \{i_1, \dots, i_n\}).(\mathcal{D}_{\xi.i_1}, \dots, \mathcal{D}_{\xi.i_n})$ où les $\mathcal{D}_{\xi.i_k}$ sont exacts.
- $(-, \xi, \{i_1, \dots, i_n\}).\mathcal{D}$ où \mathcal{D} est exact et les $\xi.i_k$ apparaissent dans \mathcal{D} .
- $(-, \xi, \{i_1, \dots, i_n\}).\mathcal{D}$ où \mathcal{D} est exact et infini.

La dernière condition indique qu'une tranche infinie exacte n'a pas besoin d'utiliser tous les lieux, elle peut se contenter de retarder à l'infini l'utilisation de lieux.

Définition 1.77 – Dessein exact

| Un dessein est **exact** si toutes ses tranches maximales pour l'inclusion sont exactes.

Remarque 1.7 Cette formulation de l'exactitude est différente de celle de Jean-Yves Girard car le but était de ne pas passer par les dessins. La figure 1.9 page ci-contre donne des exemples de desseins exacts et inexacts.

Définition 1.78 – Parcimonie

| Un dessein est **parcimonieux** si la tranche visitée durant toute normalisation avec un dessein orthogonal est exacte.

On donne la condition de gain dans le cadre linéaire (en opposition à affine) :

Définition 1.79 – Dessein gagnant

| Un dessein **gagnant** est un dessein uniforme, obstiné et parcimonieux.

Dans le cadre affine, un dessein gagnant est simplement un dessein uniforme et obstiné (sans condition de parcimonie).

Remarque 1.8 La condition de gain recherchée dans le cadre linéaire (en opposition à affine) stipule qu'un dessein est gagnant s'il est uniforme, obstiné et parcimonieux. La preuve de complétude forte [Gir01] est en fait limitée aux desseins exacts et non simplement parcimonieux.

Définition 1.80 – Dessein gagnant

| Un dessein **gagnant** est un dessein uniforme, obstiné et exact.

Proposition 1.20 – Réseau gagnant

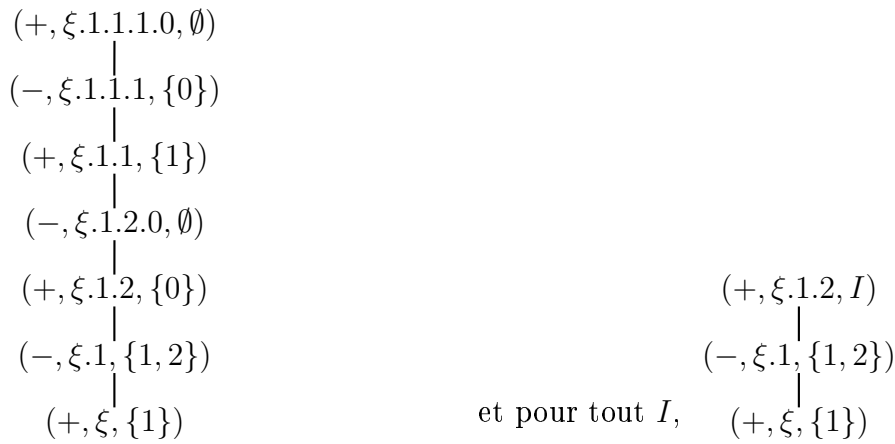
| Un réseau de desseins gagnants normalise vers un dessein gagnant.

Les chroniques infinies

Les chroniques infinies sont des desseins exacts.

Les desseins dont toutes les chroniques sont infinies sont exacts.

Une chronique finie exacte et une famille de chroniques inexactes



Un dessin exact et une variante inexacte

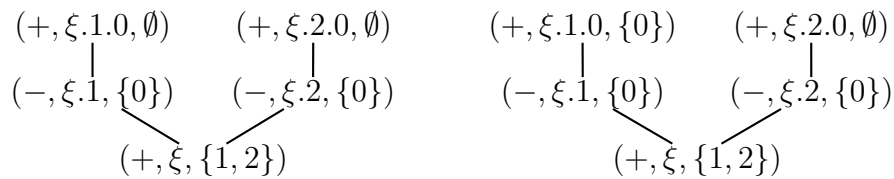


FIG. 1.9 – Desseins exacts et inexacts

1.10 La logique multiplicative additive MALL_2

Le théorème de complétude est montré pour une version de MALL_2 et sa version affine MAAL_2 . Ces calculs des séquents utilisent uniquement des formules positives (voir section 1.1 page 20) des deux côtés d'un séquent.

$$P := X \mid P \otimes P \mid P \oplus P \mid 0 \mid \exists X.P \mid \downarrow P^\perp$$

Un séquent est de la forme $\Gamma \vdash \Delta; \Sigma$. L'ensemble Σ est appelé le *bénitier* et contient au maximum une formule. Dans le cas où Σ est un singleton, Γ ne contient que des variables propositionnelles.

axiome	$\overline{X \vdash; X}$	
coupures	$\frac{\Gamma \vdash \Delta; P \quad \Gamma', P \vdash \Delta'; \Sigma}{\vdash \Gamma, \Delta; \Sigma}$	$\frac{\Gamma \vdash \Delta, P; \quad \Gamma', P \vdash \Delta';}{\vdash \Gamma, \Delta;}$
focalisation	$\frac{\Gamma \vdash \Delta; P}{\Gamma \vdash \Delta, P;}$	
multiplicatifs	$\frac{\Gamma \vdash \Delta; P \quad \Gamma' \vdash \Delta'; Q}{\Gamma, \Gamma' \vdash \Delta, \Delta'; P \otimes Q}$	$\frac{\Gamma, P, Q \vdash \Delta;}{\Gamma, P \otimes Q \vdash \Delta;}$
additifs	$\frac{\Gamma \vdash \Delta; P}{\Gamma \vdash \Delta; P \oplus Q}$	$\frac{\Gamma, P \vdash \Delta; \quad \Gamma, Q \vdash \Delta;}{\Gamma, P \oplus Q \vdash \Delta;}$
constantes	$\overline{\Gamma, 0 \vdash \Delta;}$	
quantificateurs	$\frac{\Gamma \vdash \Delta; P[A/X]}{\Gamma \vdash \Delta; \exists X.P}$	$\frac{\Gamma, P \vdash \Delta;}{\Gamma, \exists X.P \vdash \Delta;} \quad X \notin \text{Fv}(\Gamma, \Delta)$
décalages	$\frac{\Gamma, P \vdash \Delta;}{\Gamma \vdash \Delta; \downarrow P^\perp}$	$\frac{\Gamma \vdash \Delta; P}{\Gamma, \downarrow P^\perp \vdash \Delta;}$

FIG. 1.10 – MALL_2

Définition 1.81 – MALL_2

Les **règles** du calcul des séquents MALL_2 sont données figure 1.10.

Définition 1.82 – MAAL₂

Les **règles** du calcul des séquents MAAL₂ sont celles de MALL₂ (figure 1.10 page précédente) à part l'axiome qui est remplacé par : $\overline{\Gamma, X \vdash \Delta; X}$

Pour établir les théorèmes de correction et de complétude, on retire les ramifications vides pour les quantificateurs universels (on ne considère plus que des actions sous la forme $(+, \xi, I)$ avec $I \neq \emptyset$).

Soit ϕ la traduction de MALL₂ (resp. ou MAAL₂) dans la ludique (qu'on ne détaille pas).

Proposition 1.21 – Correction

Si π est une preuve d'une formule close A alors $\phi(\pi) \in \phi(A)$ et $\phi(\pi)$ est gagnant et bi-matériel.

Définition 1.83 – Formule Π_1

Une formule est Π_1 si tous ses quantificateurs sont des quantificateurs universels.

Proposition 1.22 – Complétude

Si A est une formule close Π_1 et $\mathfrak{D} \in \phi(A)$ est gagnant et bi-matériel alors il existe une preuve π de A telle que $\mathfrak{D} = \phi(\pi)$.

Il est hors de propos de montrer ce théorème ici mais l'idée de la preuve est utile. À partir d'un dessein gagnant, on essaie de retrouver la dernière règle utilisée et les conditions de gain pour cet état. Ainsi, pour le *tenseur*, la proposition 1.18 page 46 permet de remonter les hypothèses...

1.11 Tranches et ensembles d'actions

La définition suivante est originale mais la notion est standard...

Définition 1.84 – Tranchette

Une **tranchette** est l'ensemble des actions apparaissant dans une tranche.
Une **tranchette propre** est l'ensemble des actions propres apparaissant dans une tranche.

On peut associer à tout dessein \mathfrak{D} l'ensemble de ses tranches $\mathcal{ET}(\mathfrak{D})$, l'ensemble de ses tranchettes $\mathcal{ET}_\downarrow(\mathfrak{D})$ et l'ensemble de ses tranchettes propres $\mathcal{ET}_\downarrow^{pr}(\mathfrak{D})$

Proposition 1.23 – Injectivité des tranches et tranchettes

Les applications $\mathfrak{D} \mapsto \mathcal{ET}(\mathfrak{D})$, $\mathfrak{D} \mapsto \mathcal{ET}_\downarrow(\mathfrak{D})$ et $\mathfrak{D} \mapsto \mathcal{ET}_\downarrow^{pr}(\mathfrak{D})$ sont injectives.

Démonstration : Cela se prouve inductivement à l'aide des ensembles de chroniques. ■

Proposition 1.24 – Tranchettes propres d'un dessein

- L'ensemble des tranchettes propres d'un dessein (éventuellement partiel) positif peut être :
 - soit \emptyset : c'est le cas de la divergence $\mathfrak{F}i\partial$;
 - soit $\{\emptyset\}$: c'est le cas du démon ;
 - soit autre : c'est le cas des autres desseins positifs.
- La tranchette vide \emptyset est une tranchette propre de tout dessein négatif.

Chapitre 2

Contributions à la ludique simple

Sommaire

2.1	Un problème du tenseur et une solution	54
2.2	Connecteurs	58
2.3	Fax et délocalisation	59
2.4	Ordres	60
2.5	Théorèmes analytiques	60
2.6	Comportements	61
2.7	Tranchettes	61
2.8	Pour la suite	62

Ce chapitre est le premier de la thèse dans lequel sont exposées des choses nouvelles par rapport à [Gir01].

Les définitions de la ludique simple sont adaptées avec l'introduction de la divergence négative et des démons focalisés. L'ajout de ces nouveaux desseins complète au sens des tranchettes les desseins standards [Gir01].

Cette complétion pourrait paraître anecdotique pour la ludique simple mais elle est indispensable pour la suite (chapitre 3 page 65). Les *corrections* apportées ici à la ludique simple sont directement inspirées de la nécessité de définir un *tenseur* pour la ludique probabiliste.

2.1 Un problème du tenseur et une solution

Pour tout dessein négatif \mathfrak{F} de base $\xi \vdash$ et tous desseins positifs totaux et étrangers \mathcal{D} et \mathcal{E} de base $\vdash \xi$, l'adjonction du *tenseur* (définition 1.53 page 42) stipule que :

$$\llbracket \mathfrak{F}, \mathcal{D} \otimes \mathcal{E} \rrbracket = \llbracket \mathfrak{F}(\mathcal{D}), \mathcal{E} \rrbracket = \llbracket \mathfrak{F}(\mathcal{E}), \mathcal{D} \rrbracket$$

Ces équations ne sont pas généralisables au dessein partiel $\mathfrak{F}\text{id}$ comme le montre l'exemple du *tenseur* $\mathcal{D}\text{ai} \otimes \mathfrak{F}\text{id}$ car on obtiendrait dans ce cas :

$$\llbracket \mathfrak{F}, \mathcal{D}\text{ai} \otimes \mathfrak{F}\text{id} \rrbracket = \mathfrak{F}\text{id} = \mathcal{D}\text{ai}$$

Ce problème amène à définir un nouveau dessein, la divergence négative notée $\mathfrak{F}\text{id}^-$, et à poser :

$$\forall \mathcal{D}, \llbracket \mathcal{D}, \mathfrak{F}\text{id}^- \rrbracket = \mathfrak{F}\text{id}$$

En posant $\mathcal{D}\text{ai} \otimes \mathfrak{F}\text{id} = \mathfrak{F}\text{id}$ et $\mathfrak{F}(\mathfrak{F}\text{id}) = \mathfrak{F}\text{id}^-$, le problème de départ se résout et l'adjonction devient :

$$\llbracket \mathfrak{F}, \mathcal{D}\text{ai} \otimes \mathfrak{F}\text{id} \rrbracket = \llbracket \mathfrak{F}(\mathcal{D}\text{ai}), \mathfrak{F}\text{id} \rrbracket = \mathfrak{F}\text{id} = \llbracket \mathfrak{F}\text{id}^-, \mathcal{D}\text{ai} \rrbracket$$

Dorénavant, la normalisation entre le dessein positif \mathcal{D} et le dessein négatif \mathcal{E} sur des bases duales est définie ainsi :

- Si \mathcal{E} est la divergence négative, la normalisation diverge.
- Sinon, cela se passe comme en ludique simple.

L'extension de la normalisation présentée jusqu'ici ne prend pas en compte le cadre où les bases sont multiples (non unaires). En effet, l'exemple suivant montre qu'on ne sait plus définir la normalisation en présence de la divergence négative sur les bases multiples.

Soit \mathcal{D} le dessein sur la base $\vdash \xi_1, \xi_2$ formé du démon.
 Soit \mathcal{E}_1 un dessein sur la base $\xi_1 \vdash$ et \mathcal{E}_2 un dessein sur la base $\xi_2 \vdash$.
 Le résultat de $\llbracket \mathcal{D}, \mathcal{E}_1, \mathcal{E}_2 \rrbracket$ n'est pas encore défini si \mathcal{E}_1 est la divergence négative $\mathfrak{F}\text{id}^-$ et \mathcal{E}_2 ne l'est pas.

Une solution au problème qui vient d'être exposé est de bien différencier le dessein formé uniquement du démon des autres desseins et de définir des *démons focalisés* pour le remplacer dans le cas des desseins sans action propre. Le dessein formé du démon focalisé \mathfrak{F}_ξ est un nouveau dessein au même titre que la divergence négative et doit être compris comme un démon lancé à *condition* qu'on focalise sur ξ :

Définition 2.1 – Démon focalisé

Un **démon focalisé** est un démon auquel est adjoint un lieu ξ . On le note \mathfrak{X}_ξ .

Ainsi, on considère maintenant une famille plus importante d'actions :

Définition 2.2 – Actions

Les **actions** sont de la forme :

- Les actions propres : les actions (\pm, ξ, I) .
- Les actions impropres : la divergence négative \mathfrak{Fid}^- , le démon \mathfrak{X} et les démons focalisés \mathfrak{X}_ξ .

La notion d'arbre simple (définition 1.13 page 29) est redéfinie.

Définition 2.3 – Arbre simple

Un **arbre simple** \mathcal{D} est un arbre fini ou infini engendré par la grammaire suivante :

$$\begin{aligned} \mathcal{D} &:= \mathcal{D}^+ \mid \mathcal{D}_\xi^- \\ \mathcal{D}^+ &:= \mathfrak{Fid} \mid \mathcal{D}_t^+ \\ \mathcal{D}_t^+ &:= \mathfrak{X} \mid \mathfrak{X}_\xi \mid (+, \xi, \{i_1, \dots, i_n\}).(\mathcal{D}_{\xi.i_1}^-, \dots, \mathcal{D}_{\xi.i_n}^-) \\ \mathcal{D}_\xi^- &:= \mathfrak{Fid}^- \mid ((-, \xi, I). \mathcal{D}_t^+)_{I \in \mathbb{N}} \end{aligned}$$

où \mathbb{N} est un ensemble de sous-parties finies de \mathbb{N} .

Un arbre simple est **positif** s'il est de la forme \mathcal{D}^+ et **négatif**, s'il est de la forme \mathcal{D}_ξ^- .

La divergence négative sera notée explicitement. L'absence d'action négative représentant *a priori* le sconse (le dessein négatif sans actions). Ainsi, on notera

$$(+, \xi, \{i\})$$

à la place de

$$\begin{array}{c} \mathfrak{X} \mathfrak{X} \mathfrak{X} \\ | \\ (+, \xi, \{i\}) \end{array}$$

alors que \mathfrak{Fid}^- sera explicitement écrite :

$$\begin{array}{c} \mathfrak{Fid}^- \\ | \\ (+, \xi, \{i\}) \end{array}$$

Définition 2.4 – Justification

Une suite alternée d'actions est **justifiée** si pour tout préfixe \mathbf{c} de la suite se terminant par une action K alors

- ou bien K est le démon ou la divergence négative;
- ou bien K est un démon focalisé \blackbox_{ξ} et ξ^+ est dans la base β ;
- ou bien K est un démon focalisé $\blackbox_{\xi,i}$ et il y a une action $(-, \xi, J)$ dans \mathbf{c} avec $i \in J$;
- ou bien l'action K s'écrit (ϵ, ξ, I) et ξ^{ϵ} est dans la base β ;
- ou bien l'action K s'écrit (ϵ, ξ, i, I) et il y a une action $(-\epsilon, \xi, J)$ dans \mathbf{c} avec $i \in J$.

Définition 2.5 – Pré-chronique simple

Une **pré-chronique simple** sur une base β est une suite d'actions simples de polarités alternées, justifiée, sans démon (focalisé ou non) strictement à l'intérieur et telle que chaque lieu apparaît au plus une fois dans la suite.

Une pré-chronique est **positive** (resp. **négative**) si elle se termine par une action positive (resp. négative).

Définition 2.6 – Chronique

Une **chronique** est soit une pré-chronique simple positive, soit une pré-chronique simple négative se terminant par \mathfrak{Fid}^- .

On note Chron_{β} l'ensemble des chroniques sur une base β .

Définition 2.7 – Tranche

Une **tranche** est un arbre simple dans lequel l'ensemble \mathbf{N} de chaque $\mathcal{D}_{\xi}^- \neq \mathfrak{Fid}^-$ est réduit à au plus un élément.

Définition 2.8 – Linéarité des tranches

Une tranche est **linéaire** si chaque lieu d'une action de la tranche apparaît au plus une fois dans la tranche.

Remarque 2.1 En particulier, dans une tranche linéaire, s'il y a un démon focalisé \blackbox_{ξ} alors, c'est la seule action sur le lieu ξ .

Définition 2.9 – Dessein

Un arbre simple **vide** est un arbre simple qui ne contient pas d'actions différentes de \mathfrak{Fid}^- .

Un **dessein total** sur une base β est un arbre simple, non vide si β est positive, dont les branches sont des chroniques sur β et dont les tranches sont linéaires.

Définition 2.10 – Normalisation

La **réduction pas à pas** d'un réseau clos \mathfrak{R} est réécrite en :

$$\begin{array}{lcl}
 K^+ \cdot (\mathfrak{D}_1^- \dots \mathfrak{D}_n^-), (K^- \cdot \mathfrak{D}^+ \dots), \mathfrak{R} & \rightsquigarrow & \mathfrak{D}^+, \mathfrak{D}_1^-, \dots, \mathfrak{D}_n^-, \mathfrak{R} \\
 \mathfrak{X}_\xi, \mathfrak{D}_\xi^-, \mathfrak{R} & \rightsquigarrow & \mathfrak{X} \text{ si } \mathfrak{D}_\xi^- \neq \mathfrak{Fid}^- \\
 \mathfrak{X}, \mathfrak{R} & \rightsquigarrow & \mathfrak{X} \\
 \mathfrak{R} & \rightsquigarrow & \mathfrak{Fid} \\
 & & \text{dans les autres cas}
 \end{array}$$

La **normalisation** $\llbracket \mathfrak{R} \rrbracket$ d'un réseau clos \mathfrak{R} est définie par

$$\llbracket \mathfrak{R} \rrbracket = \begin{cases} \mathfrak{X} & \text{si } \mathfrak{R} \rightsquigarrow^* \mathfrak{X} \\ \mathfrak{Fid} & \text{sinon} \end{cases}$$

Ce qui se résume en :

- Le démon focalisé \mathfrak{X}_ξ coupé avec la divergence négative de base $\xi \vdash \Lambda$ diverge.
- Le démon focalisé \mathfrak{X}_ξ coupé avec un autre dessein négatif de base $\xi \vdash \Lambda$ converge (lance le démon).
- Le démon focalisé \mathfrak{X}_ξ sur la base $\vdash \xi, \Lambda_1, \xi'$ coupé avec un dessein négatif de base $\xi' \vdash \Lambda_2$ est récrit en \mathfrak{X}_ξ sur la base $\vdash \xi, \Lambda_1, \Lambda_2$.
- Les autres cas sont inchangés.

L'intérêt du remplacement des démons à la racine des desseins par des démons focalisés est de bien différencier une erreur qui sera propagée (démon focalisé) d'une erreur qui a déjà eu lieu (démon non focalisé). Le démon focalisé \mathfrak{X}_ξ à la racine d'un dessein ne sera lancé que si le dessein est coupé avec un dessein total (différent de la divergence négative).

La notion de démon focalisé résout le problème de la normalisation en base multiple en présence de divergence négative. Ainsi, la normalisation sur une base multiple est définie par :

Soit \mathfrak{D} le dessein sur la base $\vdash \xi_1, \xi_2$ formé du démon focalisé \mathfrak{X}_{ξ_i} .
 Soit \mathfrak{E}_1 un dessein sur la base $\xi_1 \vdash$ et \mathfrak{E}_2 un dessein sur la base $\xi_2 \vdash$.
 On a :

- $\llbracket \mathfrak{D}, \mathfrak{E}_1, \mathfrak{E}_2 \rrbracket = \mathfrak{X}$ si \mathfrak{E}_i n'est pas la divergence négative,
- $\llbracket \mathfrak{D}, \mathfrak{E}_1, \mathfrak{E}_2 \rrbracket = \mathfrak{Fid}$ sinon.

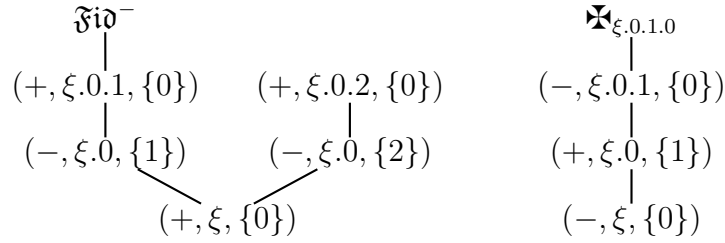
Naturellement, on garde la notion de réseau convergent :

Définition 2.11 – Réseau convergent

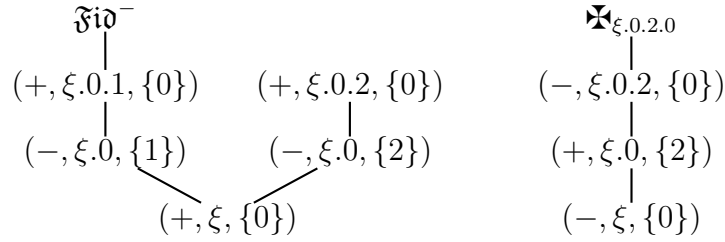
Un réseau \mathfrak{R} est **convergent** lorsque $\llbracket \mathfrak{R} \rrbracket = \mathfrak{X}$.

Quelques exemples de réseaux convergents ou non sont utiles :

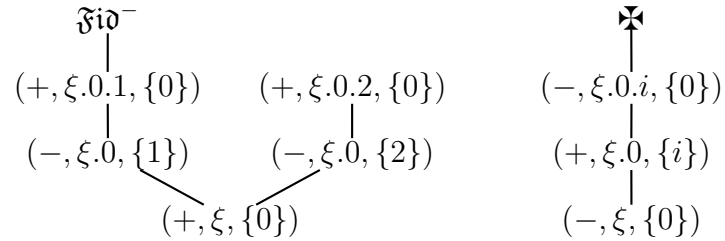
Le réseau suivant diverge :



Le réseau suivant converge :



Le réseau suivant (pour $i \in \{1; 2\}$) converge :



2.2 Connecteurs

Les connecteurs peuvent maintenant agir sur les nouveaux desseins.

Définition 2.12 – Le *tenseur*

Le tenseur est défini sur les démons focalisés et la divergence par

$$\begin{aligned}
 \mathfrak{D}^+ \otimes \mathfrak{Fid} &= \mathfrak{Fid} \\
 \mathfrak{X}_{\xi} \otimes \mathfrak{D}^+ &= \mathfrak{X}_{\xi} \quad \text{si } \mathfrak{D}^+ \text{ n'est pas } \mathfrak{Fid}
 \end{aligned}$$

Sur les autres desseins, il est défini comme à la définition 1.52 page 42.

On ne définit pas le tenseur entre le démon et la divergence car, comme cela a été montré en début de chapitre, il n'y a pas d'adjonction à gauche et à droite naturelle. Cependant, il est possible de définir un tenseur gauche (*resp.* droit) pour lequel \mathfrak{X} et \mathfrak{Fid} absorbent à gauche (*resp.* à droite). On retrouverait une adjonction à gauche ou à droite dans chacun de ces cas. Ce n'est pas décrit plus avant mais on retrouve un phénomène similaire en ludique probabiliste (page 147) qui est peut-être plus parlant. Naturellement, cela ne pose pas de problème pour définir le tenseur entre deux comportements : le démon se retrouve dans chacun de ces comportements.

Définition 2.13 – Le décalage

La définition du décalage s'étend aux nouveaux desseins.

– Si \mathcal{D} est un dessein positif de base $\vdash \xi.i$ alors

$$\uparrow \mathcal{D} = \begin{cases} (-, \xi, \{i\}).\mathcal{D} & \text{si } \mathcal{D} \neq \mathfrak{Fid} \\ \mathfrak{Stun}\mathfrak{k} & \text{sinon} \end{cases}$$

– Si \mathcal{D} est un dessein négatif de base $\xi.i \vdash$ alors

$$\downarrow \mathcal{D} = (+, \xi, \{i\}).\mathcal{D}$$

Définition 2.14 – Orthogonalité

Deux réseaux \mathfrak{R}_1 et \mathfrak{R}_2 sont orthogonaux lorsque $\mathfrak{R}_1, \mathfrak{R}_2$ est convergent.

2.3 Fax et délocalisation

Jusqu'ici, en ludique simple, le fax $\mathfrak{Fax}_{\xi \vdash \xi'}$ se comportait comme une délocalisation de ξ dans ξ' (page 39).

Ce n'est plus vrai avec la divergence négative et les démons focalisés. En effet, si \mathcal{D} est sur la base $\vdash \xi$, la forme normale du réseau \mathcal{D} , $\mathfrak{Fax}_{\xi \vdash \xi'}$ est le résultat :

- d'une délocalisation de ξ dans ξ' appliquée à \mathcal{D} ,
- suivie du remplacement de la divergence négative \mathfrak{Fid}^- par $\mathfrak{Stun}\mathfrak{k}$ et des démons focalisés par le démon (non focalisé).

Cela ouvre la porte à un nouvel objet pour la ludique pour représenter l'axiome. En effet, il serait séparé du fax car il implémenterait exactement une délocalisation. On ne poursuit pas plus avant ces questions mais cela devra être étudié. En particulier, avec un tel objet, on pourrait restreindre les desseins aux desseins de profondeur finie.

Incidemment, un phénomène similaire (l'axiome est différent du fax) apparaît dans le travail récent sur les rapports entre la ludique et la mécanique quantique de Jean-Yves Girard : le "fax" choisit une "direction" (à la manière des réseaux polarisants en optique) et de l'information est perdue dans une direction orthogonale.

Cela ne change pas pour les théorèmes de complétude à condition de définir une nouvelle classe de desseins : les desseins *présents*.

Définition 2.15 – Dessein présent

Un dessein est **présent** s'il n'utilise pas la divergence négative.

On garde exactement la même traduction de MALL_2 ou MAAL_2 en ludique. Les desseins gagnants doivent vérifier en plus la condition de présence.

2.4 Ordres

Les définitions des ordres stable et extensionnels doivent être adaptées aux nouveaux desseins.

Définition 2.16 – Ordre stable

L'ordre stable est l'ordre engendré par :

– Cas positif.

$$\mathfrak{f}id \sqsubseteq \mathfrak{D}$$

– Cas négatif. si $N \subseteq M$ alors

$$((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in N} \sqsubseteq ((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in M}$$

Définition 2.17 – Ordre extensionnel

L'ordre extensionnel est l'ordre engendré par :

– Cas positif.

$$\mathfrak{f}id \preceq \mathfrak{D}^+ \quad \mathfrak{D}^+ \preceq \mathfrak{X} \quad (+, \xi, I) \cdot \mathfrak{X} \preceq \mathfrak{X}_\xi$$

– Cas négatif.

$$\mathfrak{f}id^- \preceq \mathfrak{D}^-$$

et, si $N \subseteq M$ alors

$$((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in N} \preceq ((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in M}$$

Remarque 2.2 La divergence négative n'est pas stablement inférieure à tout dessein négatif alors qu'elle l'est extensionnellement. Ceci entraîne que l'incarnation d'un dessein présent est encore un dessein présent.

2.5 Théorèmes analytiques

Les théorèmes analytiques se conservent. La seule question est pour le théorème de séparation.

L'idée est de remarquer que :

- Le dessein $\mathfrak{f}id^-$ est séparable des autres desseins négatifs sur la base $\xi \vdash \Lambda$ grâce au démon focalisé \mathfrak{X}_ξ .
- Le dessein \mathfrak{X}_ξ est séparable du démon sur la base $\vdash \xi, \Lambda$ grâce à la divergence négative $\mathfrak{f}id^-$.

Ainsi, la chronique $\mathbf{c} \cdot (+, \xi, \{i\}) \cdot \mathfrak{f}id^-$ est différentiable de $\mathbf{c} \cdot (+, \xi, \{i\})$ grâce à un opposant qui parcourt \mathbf{c} puis joue le démon focalisé \mathfrak{X}_ξ . Symétriquement, la chronique $\mathbf{c} \cdot \mathfrak{X}_\xi$ est différentiable de $\mathbf{c} \cdot \mathfrak{X}$ grâce à un opposant qui parcourt \mathbf{c} et tel que $\mathfrak{D}_\xi^- = \mathfrak{f}id^-$.

Le théorème de stabilité se conserve ce qui donne une notion d'incarnation très similaire.

2.6 Comportements

Seules les notions d'éthique et de bi-éthique doivent être adaptées.

Définition 2.18 – Éthique

Une **éthique** est un ensemble non vide E de desseins totaux sur une même base.

Remarque 2.3 La notion d'éthique impose maintenant que l'ensemble de desseins est non vide. Cela est nécessaire car sinon l'ensemble constitué de tous les desseins négatifs (y compris la divergence négative) serait l'orthogonal d'une éthique (l'éthique positive vide).

Définition 2.19 – Bi-éthique

Une **bi-éthique** sur une base β est une paire (E, \equiv) formée d'une éthique E et d'une relation d'équivalence partielle \equiv sur E^p telle que

- Cas 1. La base β est positive.
Si $\mathcal{D} \equiv \mathcal{D}'$ alors :
 - si $\mathcal{D} = \mathfrak{F}i\mathfrak{d}$ alors $\mathcal{D}' = \mathfrak{F}i\mathfrak{d}$;
 - si $\mathcal{D} = \mathfrak{X}$ alors $\mathcal{D}' = \mathfrak{X}$ et si $\mathcal{D} = \mathfrak{X}_\xi$ alors $\mathcal{D}' = \mathfrak{X}_\xi$.
- Cas 2. La base β est négative.
 $\mathfrak{S}\mathfrak{k}\mathfrak{u}\mathfrak{n}\mathfrak{k} \equiv \mathfrak{S}\mathfrak{k}\mathfrak{u}\mathfrak{n}\mathfrak{k}$ et $\mathfrak{D}ai^- \equiv \mathfrak{D}ai^-$.

2.7 Tranchettes

La notion de divergence négative peut être mise en lumière à l'aide des tranches ou plus précisément des tranchettes. On a vu (proposition 1.24 page 52) que l'ensemble des tranchettes propres d'un dessein (éventuellement partiel) positif peut être :

- soit \emptyset : c'est le cas de la divergence
- soit $\{\emptyset\}$: c'est le cas du démon
- soit autre : c'est le cas des autres desseins positifs

et pour un dessein négatif total, l'ensemble des tranches contient au moins la tranche vide (car tout dessein négatif simple total converge face au démon).

La divergence négative est donc le dessein négatif qui n'a pas de tranchette propre alors que le **sconse** $\mathfrak{S}\mathfrak{k}\mathfrak{u}\mathfrak{n}\mathfrak{k}$ est le dessein négatif qui n'a qu'une tranchette propre : la tranchette vide. Cette notion permet donc de "clore" de façon symétrique l'ensemble des desseins vus comme des ensembles de tranchettes.

Sur une base non unaire, par exemple $\vdash \xi_1, \xi_2$, la situation est similaire. L'ensemble des tranchettes propres d'un dessein est alors une paire (une composante pour ξ_1 et une pour ξ_2) :

- soit (\emptyset, \emptyset) : c'est le cas de la divergence $\mathfrak{F}i\mathfrak{d}$
- soit $(\{\emptyset\}, \emptyset)$: c'est le cas du démon focalisé en ξ_1
- soit $(\emptyset, \{\emptyset\})$: c'est le cas du démon focalisé en ξ_2
- soit autre : c'est le cas des autres desseins positifs

Remarque 2.4 *On voit sur cette énumération qu'on pourrait ajouter d'autres desseins : les desseins dont les actions propres se répartissent en $(\{\emptyset\}, \{\emptyset\})$. Ces desseins correspondent à des produits parallèles de démons focalisés sur des lieux disjoints $\mathfrak{D} = \mathfrak{X}_{\xi_1} | \mathfrak{X}_{\xi_2}$. Si \mathfrak{E}_{ξ_1} et \mathfrak{E}_{ξ_2} sont deux desseins de bases respectives $\xi_1 \vdash$ et $\xi_2 \vdash$ alors le réseau \mathfrak{D} , \mathfrak{E}_{ξ_1} , \mathfrak{E}_{ξ_2} normaliserait si et seulement si au moins un des \mathfrak{E}_{ξ_i} n'est pas la divergence négative. On ne s'occupe pas plus avant de ces définitions dans cette thèse mais un travail actuel en collaboration avec Claudia Faggian traite d'une version très générale de desseins concurrents à partir de ce genre d'idées.*

2.8 Pour la suite

On appellera desseins dans la suite :

- les desseins positifs :
 - la divergence (positive) \mathfrak{Fid} ,
 - le démon,
 - un démon focalisé \mathfrak{X}_{ξ} ,
 - un dessin positif de la forme $(+, \xi, I).\mathfrak{R}$.
- les desseins négatifs :
 - la divergence négative \mathfrak{Fid}^- ,
 - un dessin négatif de la forme $((-, \xi, I).\mathfrak{D}_I)_{I \in \mathbb{N}}$.

Les divergences positives ou négatives sont appelées **partielles**.

Deuxième partie
Ludique probabiliste

Chapitre 3

Ludique probabiliste : desseins

Sommaire

3.1	Avant la tempête	66
3.2	Objets de base	69
3.2.1	Actions	69
3.2.2	Desseins	74
3.2.3	Réseaux	81
3.3	Normalisation	82
3.3.1	Normalisation par codage	82
3.3.2	Plongement de la ludique simple	95
3.3.3	Normalisation par tranches	97
3.4	Ordres	101
3.5	Théorèmes analytiques	103
3.5.1	Séparation	103
3.5.2	Stabilité	109
3.5.3	Associativité	110
3.5.4	Monotonie	111

L'ambition de ce chapitre est de construire une version *probabiliste* de la ludique simple, c'est-à-dire de définir un cadre contenant la ludique simple qui conserverait la plupart de ses théorèmes et propriétés. Le caractère "probabiliste" de ce nouveau modèle provient de l'utilisation de scalaires dans $[0, 1]$, de sommes bornées par 1...

L'objet de la définition d'un modèle probabiliste n'est pas de définir un nouveau modèle pour la forme mais d'étendre les théorèmes de la ludique simple à un cadre plus général (la ludique probabiliste) et d'obtenir des résultats nouveaux pour les répétitions (partie III page 171).

Une première étape a consisté à remplacer les actions positives par des mesures sur les actions positives. Ceci est insuffisant pour le théorème de séparation : les mesures sont changées en des arbres de scalaires "probabilistes" dont les feuilles sont des actions positives. Cette étape ne termine pas non plus la construction. Pour l'adjonction du tenseur, et la prise en compte de la divergence négative, il faut rajouter des coefficients devant les desseins négatifs. Par décalages, ces coefficients remontent à tous les niveaux.

3.1 Avant la tempête

L'objectif de cette section est de présenter rapidement et intuitivement les objets de la ludique probabiliste afin de faciliter la lecture de la suite de ce chapitre. Les définitions formelles ne sont données qu'à partir de la section suivante.

Un dessin probabiliste positif se présente comme un arbre de scalaires (dans $[0; 1]$) dont les feuilles sont étiquetées par des actions positives. Ces actions positives sont suivies d'arbres de scalaires suivies d'actions négatives qui sont elles-mêmes suivies d'arbres de scalaires, actions positives...

Les figures 3.1, 3.2 et 3.3 décrivent respectivement des exemples de coefficients et de desseins probabilistes. Un exemple d'interaction est montré figure 3.4 page 68.

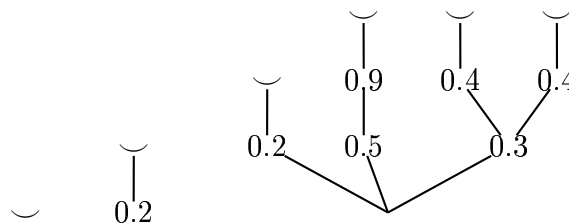


FIG. 3.1 – Exemples de coefficients

Ces desseins interagissent entre eux en collectant les arbres de scalaires suivant une mécanique similaire à celle de la ludique simple. Ultimement, sur la base vide, il ne reste plus qu'un arbre de scalaire qui est le résultat de l'interaction. Ainsi, ces arbres prennent en compte l'ordonnancement du calcul et apportent une information plus complète que pour la ludique simple.

Lors de la normalisation des deux desseins figure 3.4 page 68, on observe le coefficient suivant :

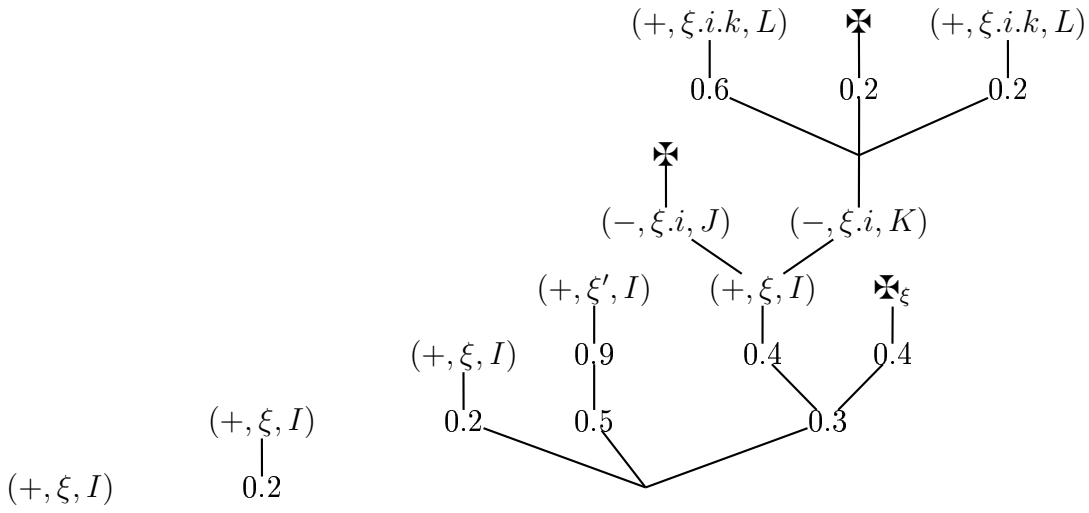


FIG. 3.2 – Exemples de desseins positifs probabilistes

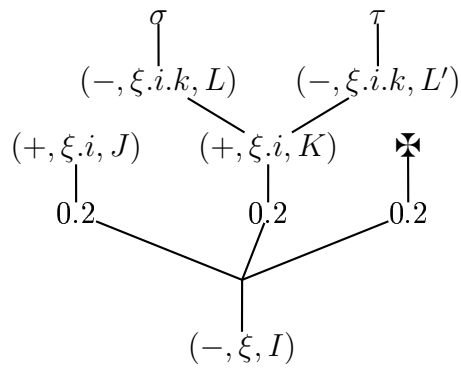


FIG. 3.3 – Un dessein négatif

La normalisation de

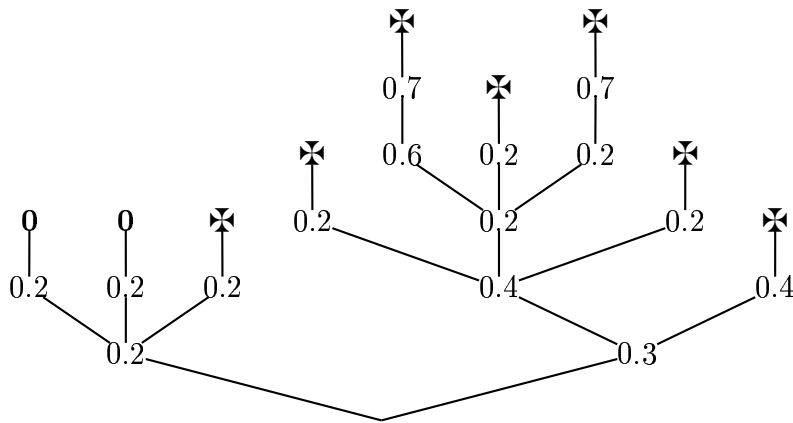
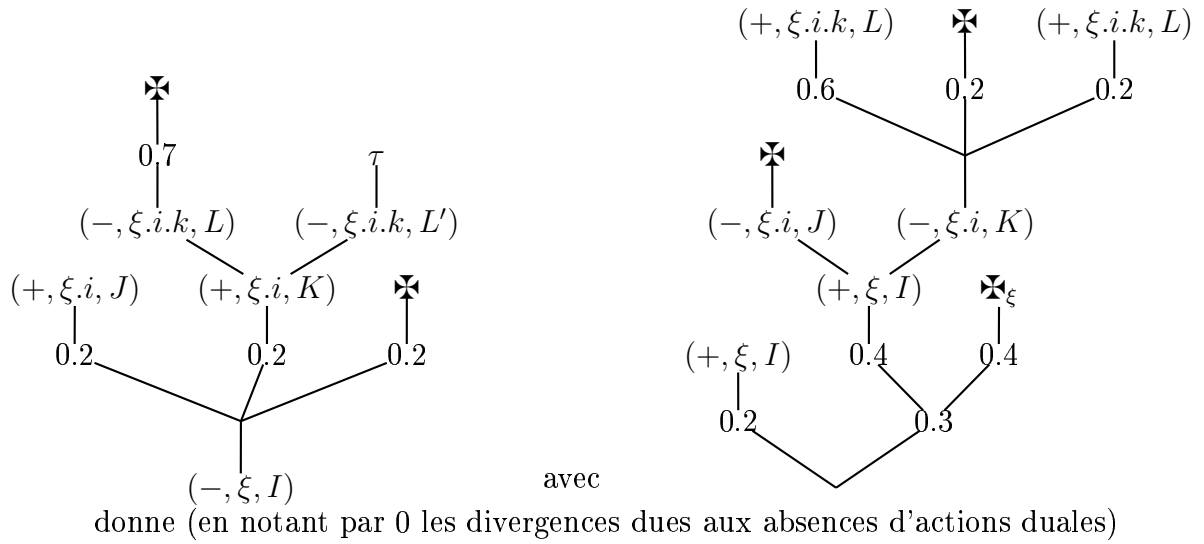
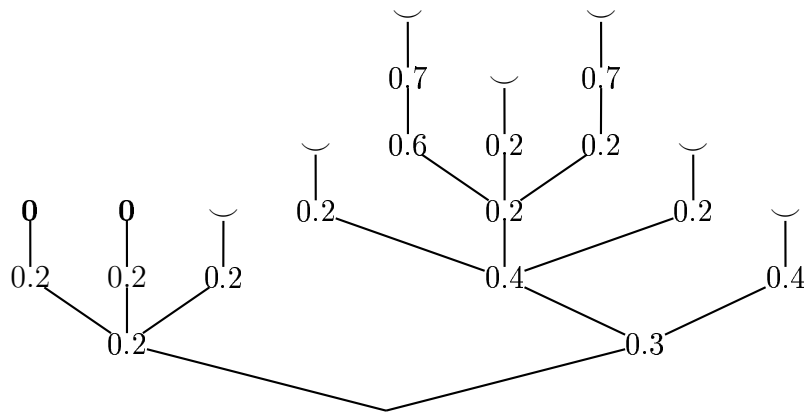


FIG. 3.4 – Une interaction



Deux desseins sont orthogonaux lorsque le poids (défini à la définition 3.7 page 71) du coefficient résultat de l'interaction est 1. Un comportement est un ensemble de desseins égal à son bi-orthogonal.

La ludique simple garde encore toute sa place à l'intérieur de ce nouveau modèle puisque les desseins simples se plongent simplement en ludique probabiliste en utilisant le coefficient réduit à une feuille. Plus fondamentalement, les comportements simples se conservent en ludique probabiliste (proposition 5.9 page 135).

3.2 Objets de base

La méthodologie utilisée pour définir les objets de bases (actions, desseins, réseaux) consiste à coller au plus près de la ludique simple tout en utilisant des coefficients.

3.2.1 Actions

Définition 3.1 – Actions simples

Les **actions** sont séparées en deux classes.

- Les actions positives sont :
 - les actions propres de la forme $(+, \xi, I)$,
 - et les actions impropres : les démons focalisés $\blacktriangleright_{\xi}$ et le démon \blacktriangleright .
- Les actions négatives sont les actions propres de la forme $(-, \xi, I)$.

Définition 3.2 – Scalaire

L'ensemble des **scalaires** est l'intervalle $[0; 1]$.

Définition 3.3 – Arbre co-fini

Un arbre est **co-fini** si de tout nœud part au moins une branche finie.

Définition 3.4 – Coefficient

Un **coefficient** fini ou infini est un arbre accepté par la grammaire :

$$c := \smile \mid \mathbf{0} \mid (s.u, \dots, s.u)$$

tel que c

- est un arbre à branchement fini
- dont les scalaires s sont dans $]0; 1[$,
- tel que à chaque branchement d'un coefficient, la somme des étiquettes du branchement est bornée par 1,
- et dont les feuilles sont de deux types : nulles $\mathbf{0}$ ou vides \smile .

Le coefficient $\mathbf{1} = \smile$ correspond au cas habituel de la ludique simple. Comme on le verra plus tard, il ne laisse pas de trace.

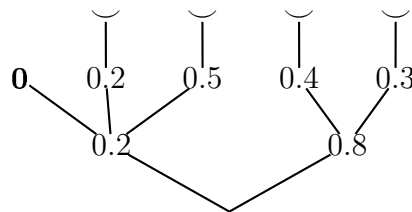
La notion d'arbre utilisée est une notion *rigide* dans le sens où les coefficients suivant sont différents :



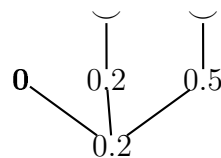
Définition 3.5 – Coefficient régulier

Un coefficient c est **régulier** si c n'utilise pas la feuille nulle $\mathbf{0}$.

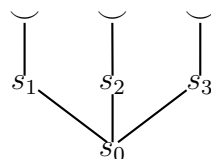
Ainsi, l'arbre suivant



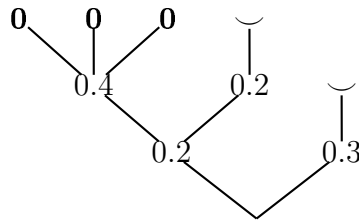
n'est pas un coefficient car le branchement



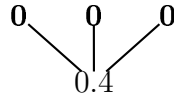
n'est pas de la forme



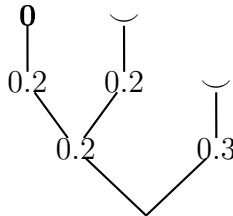
De même pour le coefficient c



à cause du bloc



Par contre, l'arbre suivant est bien un coefficient



Remarque 3.1 *On ne peut imposer que tous les coefficients soient réguliers car les coefficients du résultat de la normalisation d'un réseau \mathfrak{R} ne sont pas forcément réguliers même si tous les coefficients des desseins de \mathfrak{R} le sont.*

Définition 3.6 – Approximations d'un coefficient

Il y a deux notions d'approximation d'un coefficient.

- Un coefficient c_a est une **approximation supérieure** d'un coefficient c si c_a est obtenu en remplaçant des sous-arbres de c par la feuille vide.
- Un coefficient c_a est une **approximation inférieure** d'un coefficient c si c_a est obtenu en remplaçant des sous-arbres de c par la feuille nulle.

La remarque 3.2 page suivante montre que ces deux notions ne mènent pas à la même définition de poids. Les approximations inférieures sont retenues dans la définition suivante car elles en donnent une version croissante, ce qui n'est pas vérifié par les approximations supérieures.

Définition 3.7 – Poids d'un coefficient

Le **poids** d'un coefficient fini est défini inductivement par

$$\mathcal{W}(\mathbf{0}) = 0 \quad \mathcal{W}(\mathbf{1}) = 1 \quad \mathcal{W}((s_1.c_1, \dots, s_n.c_n)) = \sum_i s_i \mathcal{W}(c_i)$$

Le **poids** d'un coefficient infini est la borne supérieure des poids de ses approximations inférieures finies.

Proposition 3.1 – Borne pour les poids

Les poids des coefficients sont bornés par 1.

Proposition 3.2 – Monotonie des poids des approximations successives

Soit un coefficient c . Pour tout entier n , on note c_n et c'_n les approximations inférieure et supérieure de c de hauteur n .

On a :

- la suite $(\mathcal{W}(c_n))_n$ est croissante et
- la suite $(\mathcal{W}(c'_n))_n$ est décroissante.

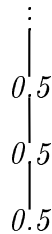
Le poids du coefficient réduit à la feuille vide **1** est 1 ce qui justifie la notation.

Remarque 3.2 La notion de co-finitude (définition 3.3 page 69) est reliée à celle de poids : tous les scalaires d'un coefficient régulier co-fini interviennent dans son poids dans le sens que la diminution de tout scalaire d'un coefficient régulier co-fini diminue la valeur de son poids. Ce n'est pas vrai pour les coefficients non réguliers comme le montre l'exemple du coefficient

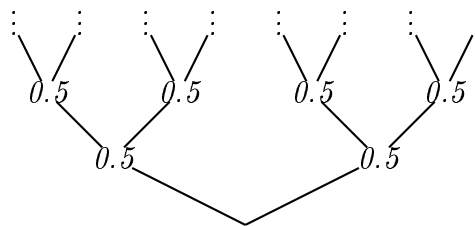


dont le poids est naturellement 0.

Ce n'est pas non plus toujours vrai pour les coefficients réguliers non co-finis comme cela est montré par l'exemple suivant

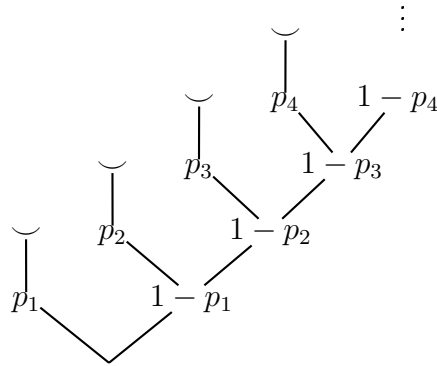


ou plus fondamentalement par le coefficient



qui est à la fois régulier, maximal (aucun scalaire ne peut être augmenté) et de poids égal à 0 (car on utilise les approximations inférieures dans la notion de poids).

Remarque 3.3 La condition de co-finitude n'est pas suffisante pour imposer une coïncidence entre la borne sup et la borne inférieure des poids de ses approximations. Ainsi, le poids d'un coefficient régulier infini n'est pas forcément la borne inférieure des poids de ses approximations supérieures finies. Un contre-exemple est le coefficient c qui décrit un arbre peigne à droite avec des coefficients sous la forme $p, (1 - p)$:



Les poids des approximations supérieures finies de c valent tous 1 mais le coefficient c n'a pas forcément un poids égal à 1 (car on utilise des approximations inférieures dans la définition du poids). Par exemple, si $\sum_i p_i < 1$ alors $\mathcal{W}(c) < \sum_i p_i < 1$. On en déduit immédiatement que si la série $\sum_i p_i$ converge, alors le poids de c est strictement inférieur à 1.

Définition 3.8 – Poids d'une branche d'un coefficient

Le **poids** d'une branche finie d'un coefficient est le produit des scalaires apparaissant dans cette branche.

Proposition 3.3 – Poids d'un coefficient et des branches

Le poids d'un coefficient c est la somme (infinie) des poids de ses branches se terminant par une feuille vide.

Démonstration : Pour un coefficient fini, c'est vrai (c'est une récurrence sur la hauteur de l'arbre en utilisant la distributivité de la multiplication sur l'addition).

Pour un coefficient infini, on procède comme suit. Le poids d'un coefficient c est la borne supérieure des poids de ses approximations inférieures finies, c'est-à-dire la borne supérieure quand n varie des sommes des poids de ses branches se terminant par une feuille vide de longueur inférieure ou égale à n . C'est exactement la définition de la somme infinie des poids de ses branches se terminant par une feuille vide. ■

Remarque 3.4 La proposition précédente explique pourquoi la notion de poids est reliée aux approximations inférieures : c'est cette définition qui donne ce résultat.

Dans la proposition suivante, on utilise la notation $c = c'.(c_1, \dots, c_n)$ pour le coefficient c' dont les feuilles vides (aux occurrences données $\alpha_1, \dots, \alpha_n$ sont remplacées par les coefficients c_i). En particulier, cette notation est aussi utilisée pour c' ayant un nombre infini de feuilles vides.

Proposition 3.4 – Coefficients de poids 1

Soit $c = (s_1.c_1, \dots, s_n.c_n)$ un coefficient. On a :

$$\mathcal{W}(c) = 1 \iff \forall i, \mathcal{W}(c_i) = 1 \wedge \sum_i s_i = 1$$

Un coefficient de poids 1 est régulier et co-fini.

De plus, diminuer un scalaire ou remplacer un sous-coefficient par 0 dans un coefficient de poids 1 diminue strictement son poids.

Enfin, un coefficient $c = c'.(c_1, \dots, c_n)$ est de poids 1 si et seulement si c' et les (c_i) sont de poids 1.

Démonstration :

1. Le premier point est une conséquence directe de la définition de poids.
2. Le deuxième point découle du fait qu'un coefficient sans nœud co-fini est de poids nul. La régularité est une induction sur le premier point.
3. Le troisième point est une conséquence directe par induction du premier point.
4. Pour le quatrième point, on prouve les deux sens :

\Rightarrow Supposons $\mathcal{W}(c) = 1$.

Le coefficient c' est une approximation supérieure de c donc $\mathcal{W}(c') = 1$.

Soit α_{i_0} une occurrence d'une feuille vide de c , montrons que c_{i_0} est de poids 1.

Soit c'_{i_0} le coefficient obtenu en remplaçant α_{i_0} par c_{i_0} mais en gardant les autres feuilles vides de c' .

Le coefficient c'_{i_0} est une approximation supérieure de c donc est de poids 1.

Soit b la branche de c' se terminant en α_{i_0} . Une branche finie se terminant par une feuille vide de c'_{i_0} est soit une branche de c' se terminant par une feuille vide différente de b soit b suivie d'une branche de c_{i_0} se terminant par une feuille vide. Donc la somme des poids des branches se terminant par une feuille vide de c'_{i_0} est égale au poids de c' moins le poids de b plus le poids de b fois la somme des poids des branches se terminant par une feuille vide de c_{i_0} . Donc $1 = \mathcal{W}(c'_{i_0}) = \mathcal{W}(c'_{i_0}) - \mathcal{W}(b) + \mathcal{W}(b) * \mathcal{W}(c_{i_0})$. Or $\mathcal{W}(b) \neq 0$ donc $\mathcal{W}(c_{i_0}) = 1$.

\Leftarrow Supposons que c' et les (c_i) sont de poids 1.

Soit $\epsilon \in]0; 1]$. Soit c'_ϵ et (c_{i_ϵ}) des approximations inférieures finies de c' et des (c_i) de poids supérieurs à $1 - \epsilon$.

Soit $c_\epsilon = c'_\epsilon.(c_{i_\epsilon})$ l'approximation inférieure finie de c qui consiste à remplacer les feuilles vides de c'_ϵ par le coefficient c_{i_ϵ} correspondant. On a $\mathcal{W}(c) \geq \mathcal{W}(c_\epsilon) \geq (1 - \epsilon)^2$ (par finitude du coefficient).

Ceci est vrai pour tout ϵ positif donc $\mathcal{W}(c) = 1$. ■

3.2.2 Desseins

Pour définir la notion de dessein (définition 3.13 page 77), il faut d'abord passer par une version sans condition de cohérence : les arbres probabilistes.

Définition 3.9 – Arbre probabiliste

Un **arbre probabiliste** \mathfrak{D} est un arbre fini ou infini engendré par la grammaire suivante :

$$\begin{array}{ll}
 \text{arbre} & \mathfrak{D} := \mathfrak{D}^+ \mid \mathfrak{D}_\xi^- \\
 \\
 \text{arbre positif} & \mathfrak{D}_t^+ := \mathbf{0} \mid \mathfrak{D}_t^+ \\
 \text{arbre positif non vide} & \mathfrak{D}_s^+ := c.(\mathfrak{D}_s^+, \dots, \mathfrak{D}_s^+) \\
 \text{arbre positif semi-simple} & \mathfrak{D}_s^+ := \mathfrak{X} \mid \mathfrak{X}_\xi \mid (+, \xi, \{i_1, \dots, i_n\}).(\mathfrak{D}_{\xi.i_1}^-, \dots, \mathfrak{D}_{\xi.i_n}^-) \\
 \text{arbre négatif} & \mathfrak{D}_\xi^- := c.(\mathfrak{D}_{s_\xi}^-, \dots, \mathfrak{D}_{s_\xi}^-) \\
 \text{arbre négatif semi-simple} & \mathfrak{D}_{s_\xi}^- := ((-, \xi, I).\mathfrak{D}_t^+)_{I \in \mathbb{N}}
 \end{array}$$

où

- \mathbb{N} est un ensemble de sous-parties finies de \mathbb{N}
- $c.(\mathfrak{D}_s^+, \dots, \mathfrak{D}_s^+)$ représente un arbre composé du coefficient c différent de $\mathbf{0}$ suivi par un dessin \mathfrak{D}_s^+ pour chaque feuille vide de c . Ce coefficient c est dit **positif**.
- $c.(\mathfrak{D}_{s_\xi}^-, \dots, \mathfrak{D}_{s_\xi}^-)$ représente un arbre composé du coefficient c (possiblement égal à $\mathbf{0}$) suivi par un dessin $\mathfrak{D}_{s_\xi}^-$ pour chaque feuille vide de c . Ce coefficient c est dit **négatif**.

Un arbre probabiliste est **positif** s'il est de la forme \mathfrak{D}^+ et **négatif**, s'il est de la forme \mathfrak{D}_ξ^- .

On a besoin d'un coefficient à la racine des arbres négatifs pour rendre compte de la divergence négative (introduite section 2.1 page 54) pour l'adjonction du *tenseur* en ludique probabiliste (proposition 5.25 page 147). Ces coefficients se retrouvent à tout étage des arbres à cause des décalages.

La condition stipulant qu'une action négative est suivie d'un coefficient non nul est nécessaire pour la séparation (proposition 3.21 page 108). Cependant, comme en ludique simple (avec la divergence \mathfrak{Fid} [Cur04b, Gir01]), on se permet, par abus de notation, de noter l'absence d'une action négative $(-, \xi, \{I\})$ par

$$\begin{array}{c}
 \mathbf{0} \\
 | \\
 (-, \xi, \{I\})
 \end{array}$$

Définition 3.10 – Les divergences \mathfrak{Fid} et \mathfrak{Fid}^-

Dans le cadre de la ludique probabiliste, l'arbre positif restreint à la feuille nulle $\mathbf{0}$ est noté \mathfrak{Fid} et l'arbre négatif restreint à la feuille nulle $\mathbf{0}$ est noté \mathfrak{Fid}^- ce qui explique l'absence d'une action spécifique \mathfrak{Fid}^- .

Définition 3.11 – Ordre d'inclusion

L'**inclusion** entre arbres probabilistes est l'ordre engendré par :

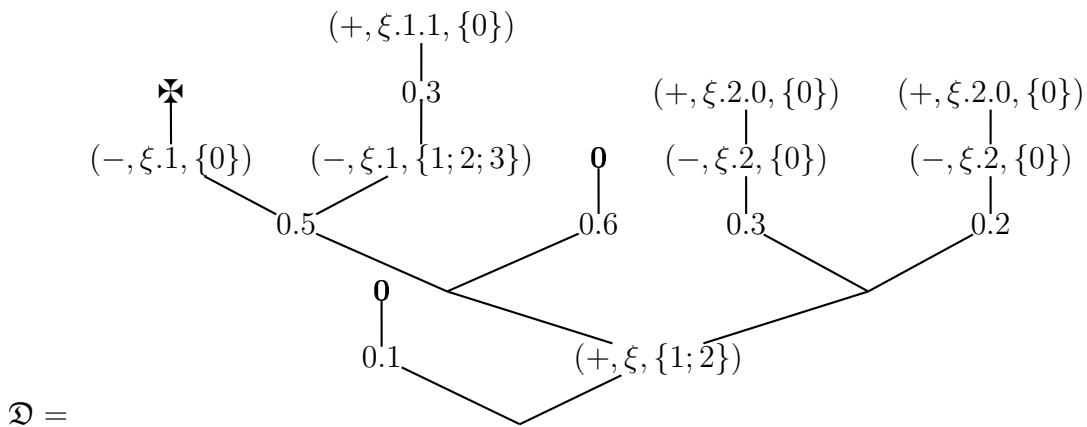
– Cas positif.

$$\mathbf{0} \subseteq \mathfrak{D}$$

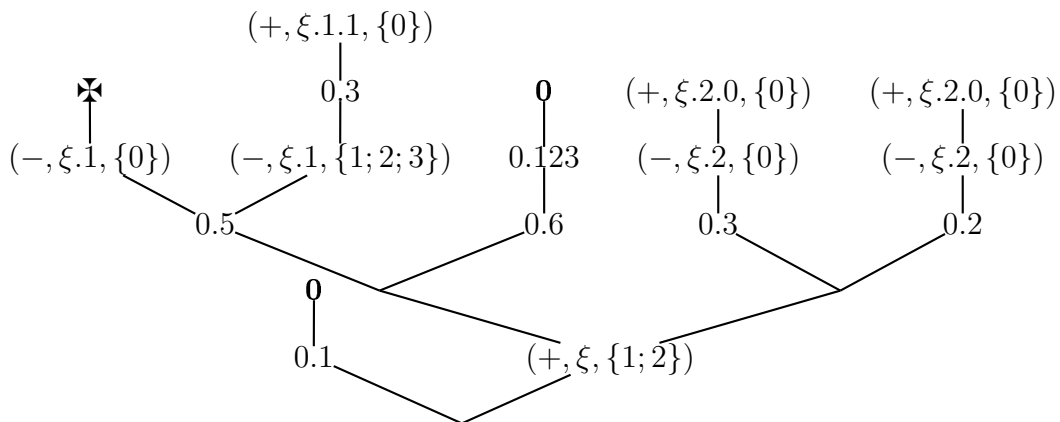
– Cas négatif. si $N \subseteq M$ alors

$$((-, \xi, I). \mathfrak{D}_I)_{I \in N} \subseteq ((-, \xi, I). \mathfrak{D}_I)_{I \in M}$$

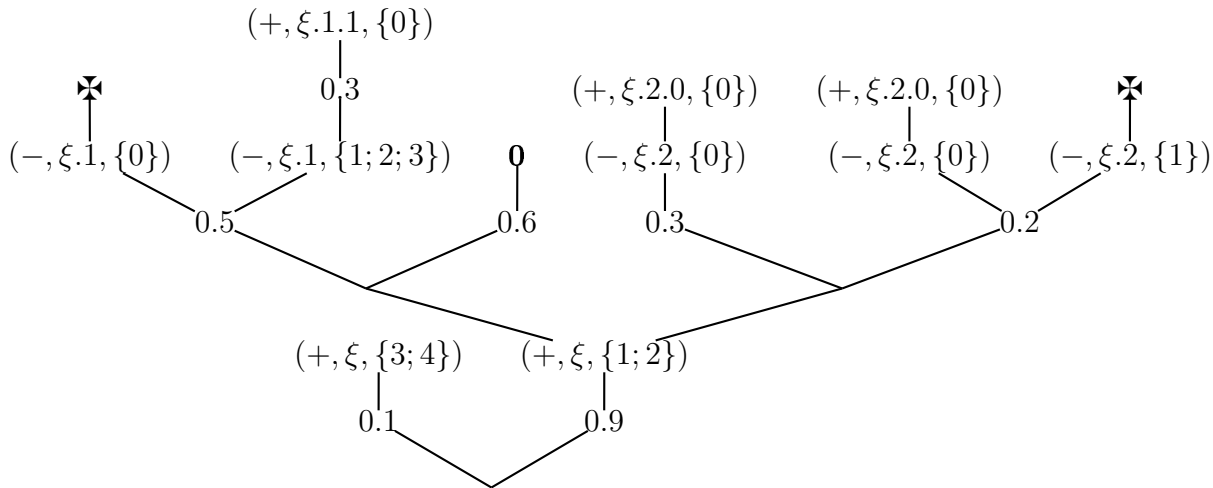
Un contre-exemple est utile pour bien comprendre l'inclusion. Le dessin



n'est pas inclus dans



car la feuille nulle $\mathbf{0}$ remplacée est négative. Par contre \mathfrak{D} est inclus dans



car le changement entre ces deux desseins est simplement l'ajout d'une nouvelle action $(-, \xi.2, \{1\})$ et de sa continuation dans le dessin négatif le plus à droite (au dessus du scalaire 0.2) ainsi que le remplacement d'une feuille nulle positive par un dessin positif.

Définition 3.12 – Projection simple

Une **projection simple** d'un arbre probabiliste est un arbre simple obtenu par remplacement de ses coefficients c (et de leurs continuations) par :

- si c est de poids nul, la divergence de la polarité donnée;
- sinon (c a des feuilles vides),
 - la continuation d'une feuille vide de c ;
 - ou éventuellement, la divergence de la polarité donnée si c est de poids différent de 1.

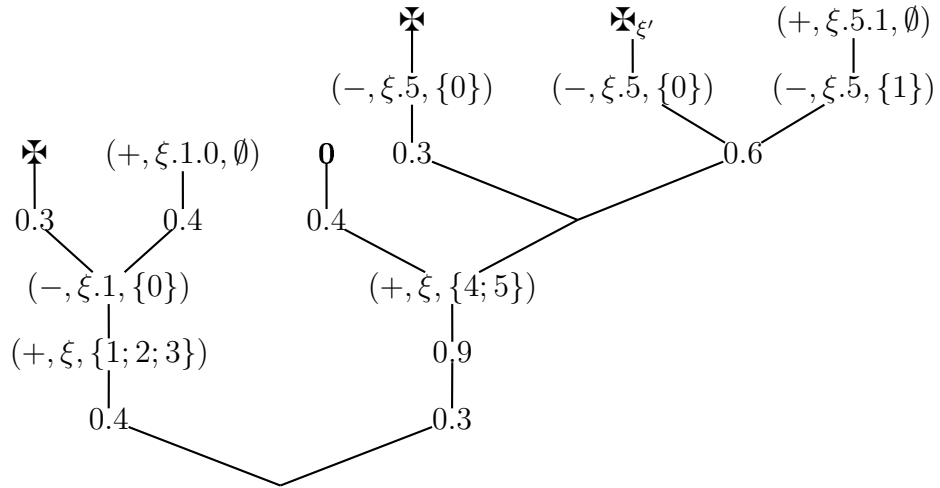
Définition 3.13 – Dessin probabiliste

Un **dessein** sur une base β est un arbre probabiliste sur β tel que ses projections simples sont des desseins simples sur la base β .

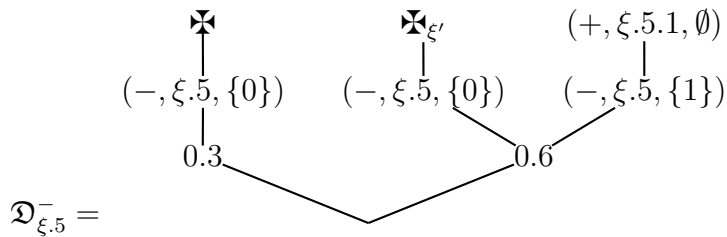
Dans la suite de la thèse, sauf précision, la notion de dessein utilisée est celle de dessein probabiliste.

Deux exemples de desseins probabilistes peuvent éclairer la définition :

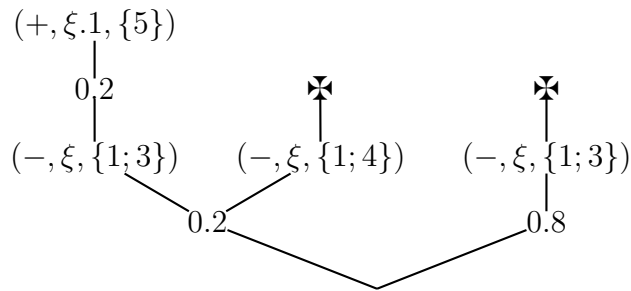
1. Un dessein positif.



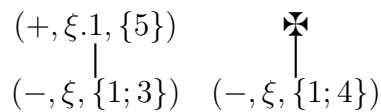
On voit que les coefficients sont utilisés à la fois avant les actions positives et les actions négatives. Le coefficient $\mathbf{0}$ est après une action positive, il correspond donc à la divergence négative. On a ici $\mathcal{D}_{\xi,4}^- = 0.4.\mathbf{0}$ et



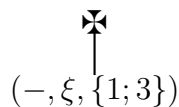
2. Un dessin négatif.



Ce dessin négatif a exactement deux projections simples : la partie de gauche

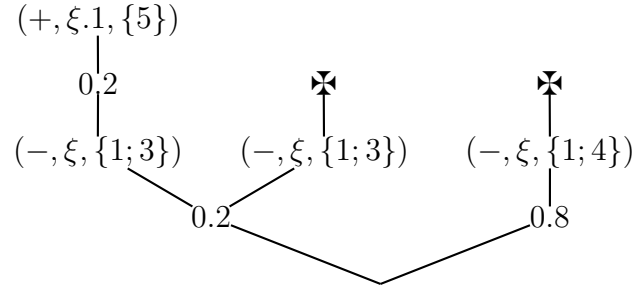


et la partie de droite



Ce qui explique la présence de deux actions $(-, \xi, \{1; 3\})$.

3. L'arbre probabiliste suivant n'est pas un dessein car il y a deux actions $(-, \xi, \{1; 3\})$ dans la même projection simple.



Le passage de la ludique simple à la ludique probabiliste en ajoutant des coefficients aux desseins amène à reconnaître des formes particulières de desseins : les desseins semi-simples, complets et les immédiatement totaux. Toutes ces notions sont importantes pour la suite.

Définition 3.14 – Dessein semi-simple

Un dessein est **semi-simple** si son coefficient à la racine est égal à 1.

Définition 3.15 – Combinaison linéaire

Soit c un coefficient et α_i les occurrences des feuilles vides de c . Le dessein dont le coefficient à la base est c et les feuilles vides sont associées aux desseins \mathcal{D}_{α_i} est la **combinaison linéaire** de coefficient c des desseins (\mathcal{D}_{α_i}) .

On note les combinaisons linéaires $c.(\mathcal{D}_{\alpha_i})$.

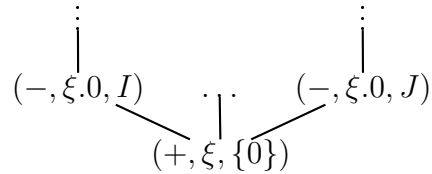
Par abus de notation, on note aussi par la suite $c.(\mathcal{D}_1, \dots, \mathcal{D}_n)$ une combinaison linéaire finie ou infinie.

Remarque 3.5 On n'impose pas que les desseins (\mathcal{D}_{α_i}) soient semi-simples dans la définition d'une combinaison linéaire. Ainsi, des coefficients peuvent être "recollés" par combinaison linéaire.

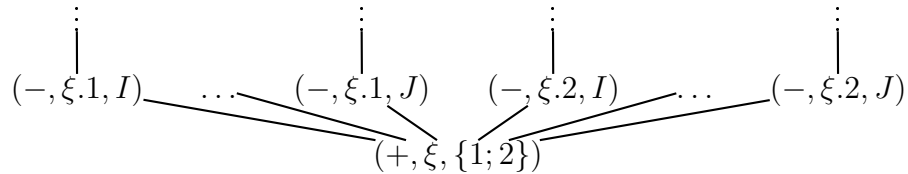
Par la suite, une syntaxe est

- $(+, \xi, I).\mathfrak{R}$, $(+, \xi, I).\mathfrak{R}_1\mathfrak{R}_2$ ou $(+, \xi, I).(\mathcal{D}_1, \dots, \mathcal{D}_n)$ pour dénoter un dessein positif semi-simple suivi d'un ensemble de desseins $\mathfrak{R} = \mathfrak{R}_1 \cup \mathfrak{R}_2 = \{\mathcal{D}_1, \dots, \mathcal{D}_n\}$.
- $((-, \xi, I).\mathcal{D}_I)_{I \subset_{fin} \mathbb{M}}$ pour dénoter un dessein négatif semi-simple.
- $c.(\mathcal{D}_1, \dots, \mathcal{D}_n)$ dénotera un dessein, positif ou négatif suivant la polarité des \mathcal{D}_i , dont le coefficient de base est c ou, dans le cas où les \mathcal{D}_i ne sont pas semi-simples, $c.(c_1, \dots, c_k)$ pour une certaine famille $(c_i)_i$.
- On utilise naturellement une syntaxe d'arbre.

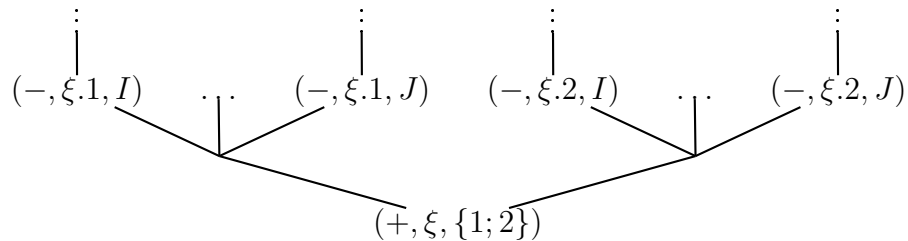
Il n'y a pas d'ambiguïté dans le cadre d'une ramification avec un seul entier comme $\{0\}$



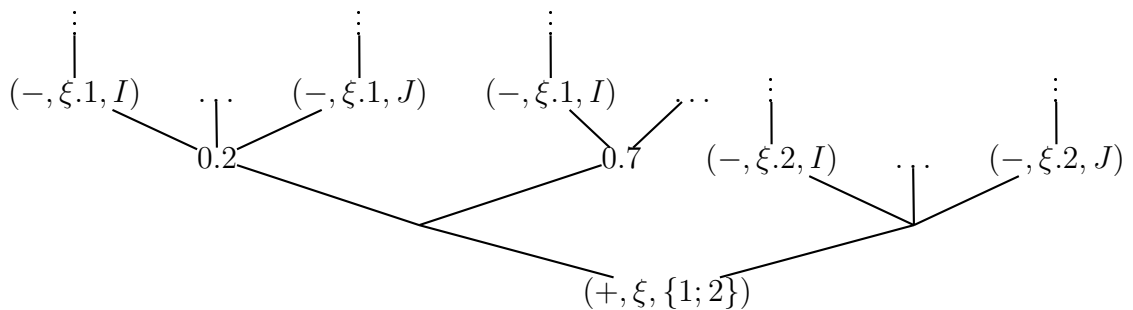
Pour une ramifications plus complexe comme $\{1; 2\}$ par exemple, on notera



ou



On procède de même avec des coefficients.



Proposition 3.5 – Combinaison linéaire

Toute combinaison linéaire de desseins sur une base donnée est un dessin de cette base.

Proposition 3.6 – Décomposition en desseins semi-simples

Tout dessin se décompose de manière unique en une combinaison linéaire de desseins semi-simples.

Définition 3.16 – Dessin complet

Un dessin différent de $\mathbf{0}$ est **complet** si tous ses coefficients sont soit nuls soit de poids 1.

Un dessin est **positivement complet** si tous ses coefficients positifs sont de poids 1.

Un dessin complet ou positivement complet peut contenir des coefficients $\mathbf{0}$ pour représenter la divergence négative.

Définition 3.17 – Dessin immédiatement total

Un dessin est *immédiatement total* si le poids de son coefficient à la racine est 1.

3.2.3 Réseaux

Comme en ludique simple, les dessins interagissent entre eux quand ils sont placés sur des lieux identiques (section 1.2.3 page 30).

Définition 3.18 – Pré-réseau

Un *pré-réseau* est un ensemble fini de dessins sur des bases β_i tel qu'un lieu apparaît au plus dans deux bases : une fois positivement et une fois négativement.

Définition 3.19 – Coupure

Une *coupure* dans un pré-réseau \mathfrak{R} est la donnée de deux dessins de \mathfrak{R} dont les bases ont un lieu en commun.

Définition 3.20 – Réseau

Un *réseau* est un pré-réseau dans lequel la relation de coupure est connexe et acyclique.

Définition 3.21 – Base de réseau

La *base* d'un réseau \mathfrak{R} est la base formée de l'ensemble des lieux (avec leur polarité) des bases des dessins de \mathfrak{R} qui n'apparaissent qu'une fois.

Définition 3.22 – Réseau clos

Un réseau est *clos* lorsque sa base est vide.

Définition 3.23 – Actions et lieux visibles

Soit \mathfrak{R} un réseau.

Un lieu est *visible* dans \mathfrak{R} si c'est un sous-lieu d'un lieu de la base de \mathfrak{R} .

Une action dans un dessin de \mathfrak{R} est *visible* si c'est le démon, un démon focalisé ou une action propre dont le lieu visible.

Définition 3.24 – Dessin principal

Le *dessin principal* d'un réseau est

- l'unique dessin positif de ce réseau s'il existe,
- ou l'unique dessin négatif de base $\xi \vdash \Lambda$ telle que ξ est visible.

3.3 Normalisation

Cette section décrit la normalisation de la ludique probabiliste. Le chapitre 4 page 113 complète son étude.

La normalisation s'opère sur les réseaux et associe à un réseau de base donnée un dessein sur la même base. Plus généralement, la normalisation d'un pré-réseau acyclique donne un pré-réseau et la connexité implique en plus l'obtention d'un dessein.

On peut définir la normalisation de différentes façons : par tranches, avec une machine abstraite et enfin par codage en ludique simple. Les versions par tranches ou codage en ludique simple sont les plus simples mais elles ne se généralisent pas au cas avec répétitions.

Comme expliqué dans la suite (définition 3.30 page 91), dans le cas clos, la normalisation d'un réseau clos donne fondamentalement un coefficient dont certaines branches sont terminées par des démons. Un réseau converge lorsque le poids de ce coefficient est à 1.

L'utilisation de coefficients sous forme d'arbre et non simplement de mesures probabilistes prend son intérêt ici : un résultat n'est pas qu'un booléen (le cas de la ludique simple) ou même un nombre réel (ce qui arriverait si on prenait des mesures) mais un arbre de réels. C'est cette géométrie qui donne le théorème de séparation à la proposition 3.21 page 108 et surtout le conservera avec des répétitions (chapitre 8 page 205).

3.3.1 Normalisation par codage

La définition la plus simple de la normalisation en ludique probabiliste utilise un *codage* en ludique simple. Pour chaque dessein de base β positive, quitte à délocaliser les lieux de la base pour ne pas utiliser le lieu vide, on définit un dessein de base $\beta \cup \{\xi_c^+\}$ où ξ_c est un nouveau lieu. Chaque coefficient est remplacé par un arbre d'actions sur des sous-lieux de ξ_c .

Pour simplifier la définition de la normalisation, celle-ci est définie en deux étapes : pour une classe restreinte de desseins (les desseins positivement probabilistes) puis pour tous les desseins.

Définition 3.25 – Dessein positivement probabiliste

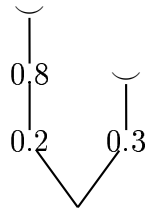
Un dessein est **positivement probabiliste** si ses coefficients négatifs sont égaux à 1.

Dans un premier temps, les desseins considérés sont supposés positivement probabilistes.

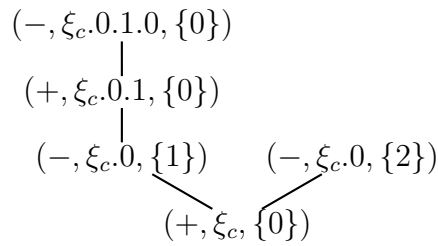
Intuitions

La traduction étant relativement technique, quelques exemples intuitifs sont présentés. Le but est ici de traduire un coefficient possédant n -feuilles vides en un arbre d'actions simples de même géométrie (avec n -feuilles négatives).

Ainsi, le coefficient $\overbrace{0.8}^{\smile}$ peut être traduit en $(-, \xi_c, 0, \{0\})$
 Le coefficient $\underbrace{0.8}_\smile$ peut être traduit en $(+, \xi_c, \{0\})$.



est traduit en

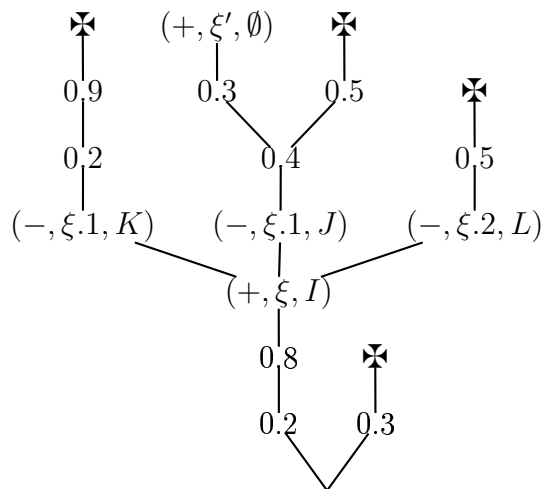


Plus généralement, un coefficient est traduit en un arbre d'actions sur le lieu ξ_c tel que

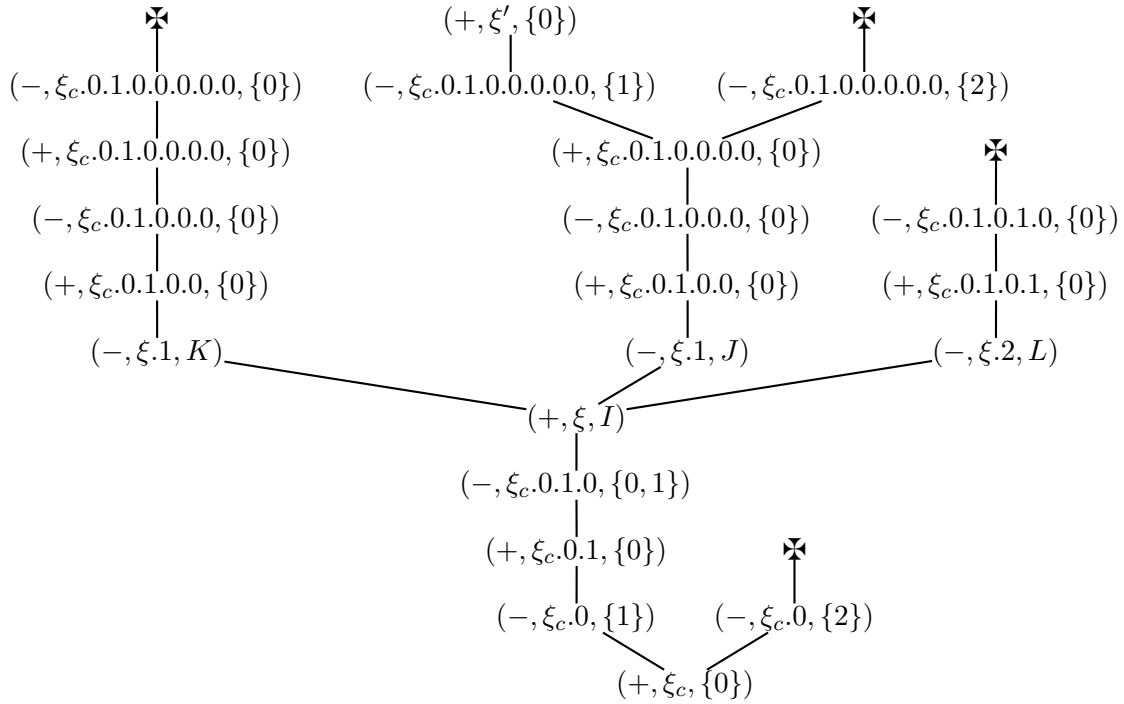
- chaque scalaire est traduit par un couple action positive/action négative ;
- en cas de branchement dans l'arbre du coefficient, les actions positives traduites sont partagées ;
- la feuille vide correspond à une feuille négative dont on définit la continuation.

Ce codage s'étend naturellement aux desseins :

Le dessin



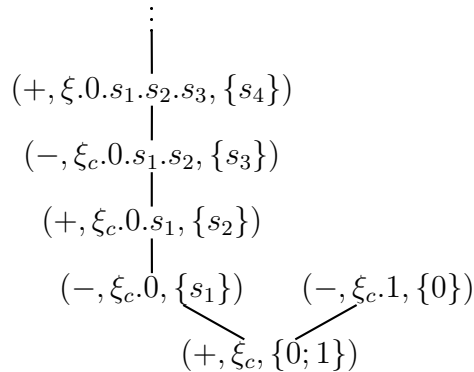
est traduit en



Le codage utilisé

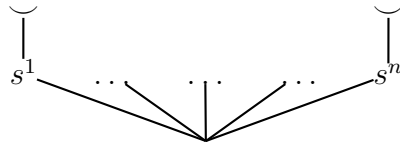
La traduction esquissée jusqu'ici traduit uniquement la géométrie des coefficients et non la valeur des scalaires. Une telle traduction (scalaires traduits par un nombre fini d'actions) est impossible pour des raisons de cardinalité. En revanche, on peut coder la valeur d'un scalaire dans un dessin simple infini.

Soit s un scalaire. Le scalaire s appartient à l'intervalle $]0; 1[$ et s'écrit donc en binaire $0.s_1s_2\dots$ (l'unicité ou non de l'écriture n'intervient pas). On traduit un coefficient par

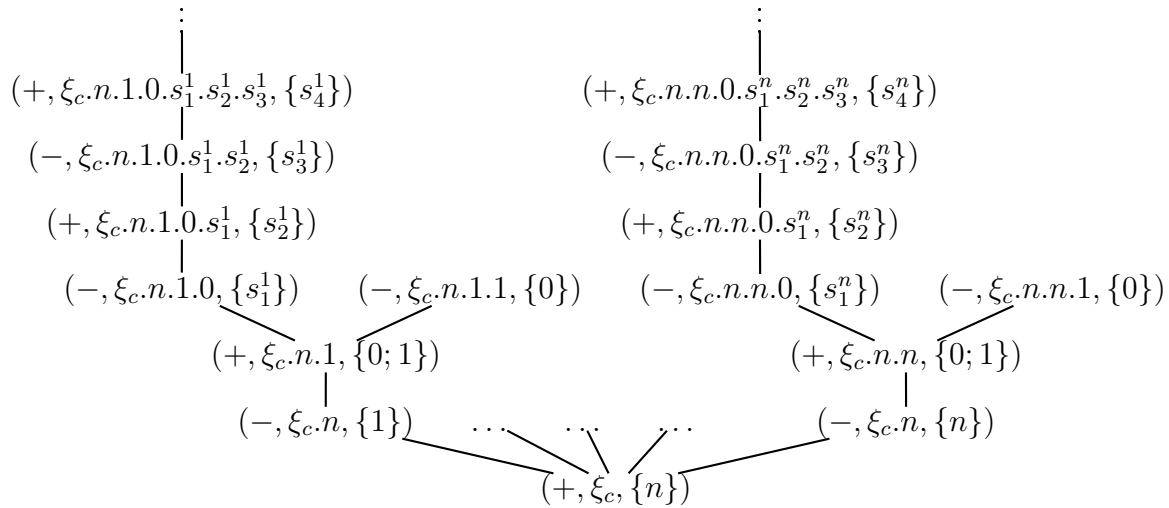


La partie de gauche code la valeur de s et la partie de droite libère simplement de la place pour un futur dessin.

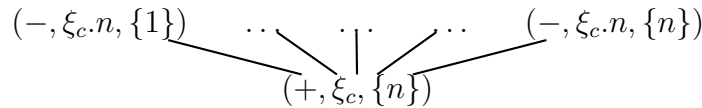
Un coefficient



où $s^i = 0.s_1^i s_2^i \dots$ est traduit par



On remarque que le branchement est fait à l'aide d'un bloc

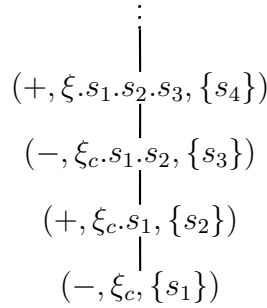


L'utilisation d'actions sur un même lieu $\xi_c.n$ est utile ici car, ces actions étant incohérentes, il n'y a aucune condition sur le partage des lieux au-dessus des actions $(-, \xi_c.n.i.1, \{0\})$. C'est nécessaire pour coder les coefficients car il n'y a pas de condition entre les continuations de deux feuilles d'un coefficient (remarque 3.6 page suivante). On utilise le lieu $\xi_c.n$ et non le lieu $\xi_c.0$ par exemple car le n donne le cardinal du branchement.

Le coefficient 0 est traduit par le dessin réduit à l'action $(+, \xi_c, \{0\})$.

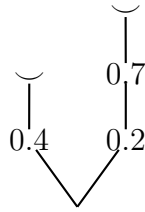
Définition 3.26 – Le dessein $\mathfrak{D}_s(\xi_c)$

Soit $s \in]0; 1[$. Le **codage** du scalaire s est le dessein $\mathfrak{D}_s(\xi_c)$:

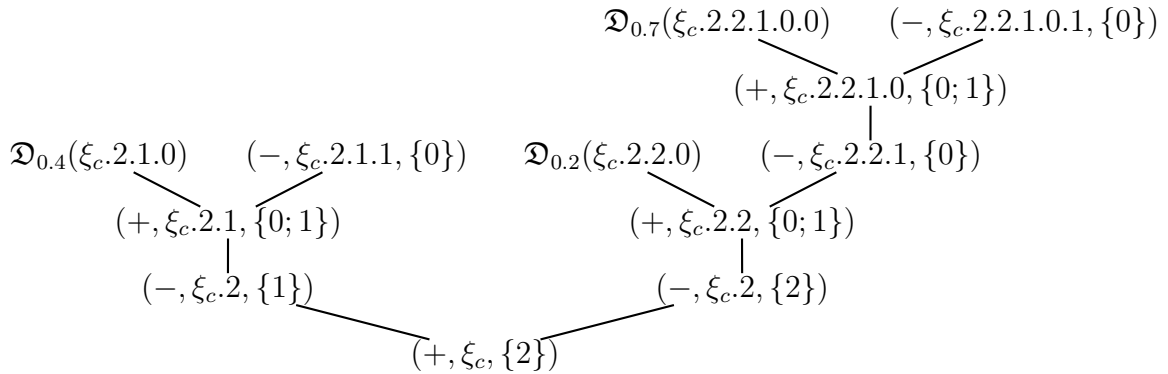


où $s = 0.s_1s_2\dots$

Les définitions esquissées jusqu'ici se composent. Ainsi, le coefficient :

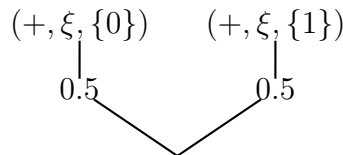


est traduit par

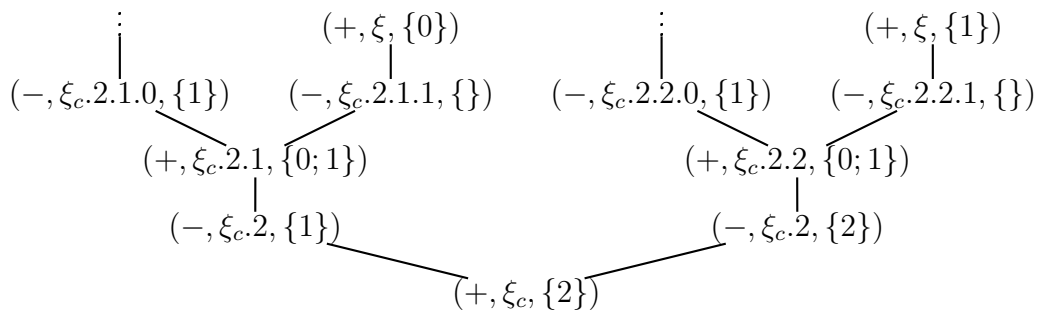


Il y a naturellement deux actions négatives maximales qui correspondent aux feuilles vides du coefficient.

Remarque 3.6 *Le dessein*



est codé en

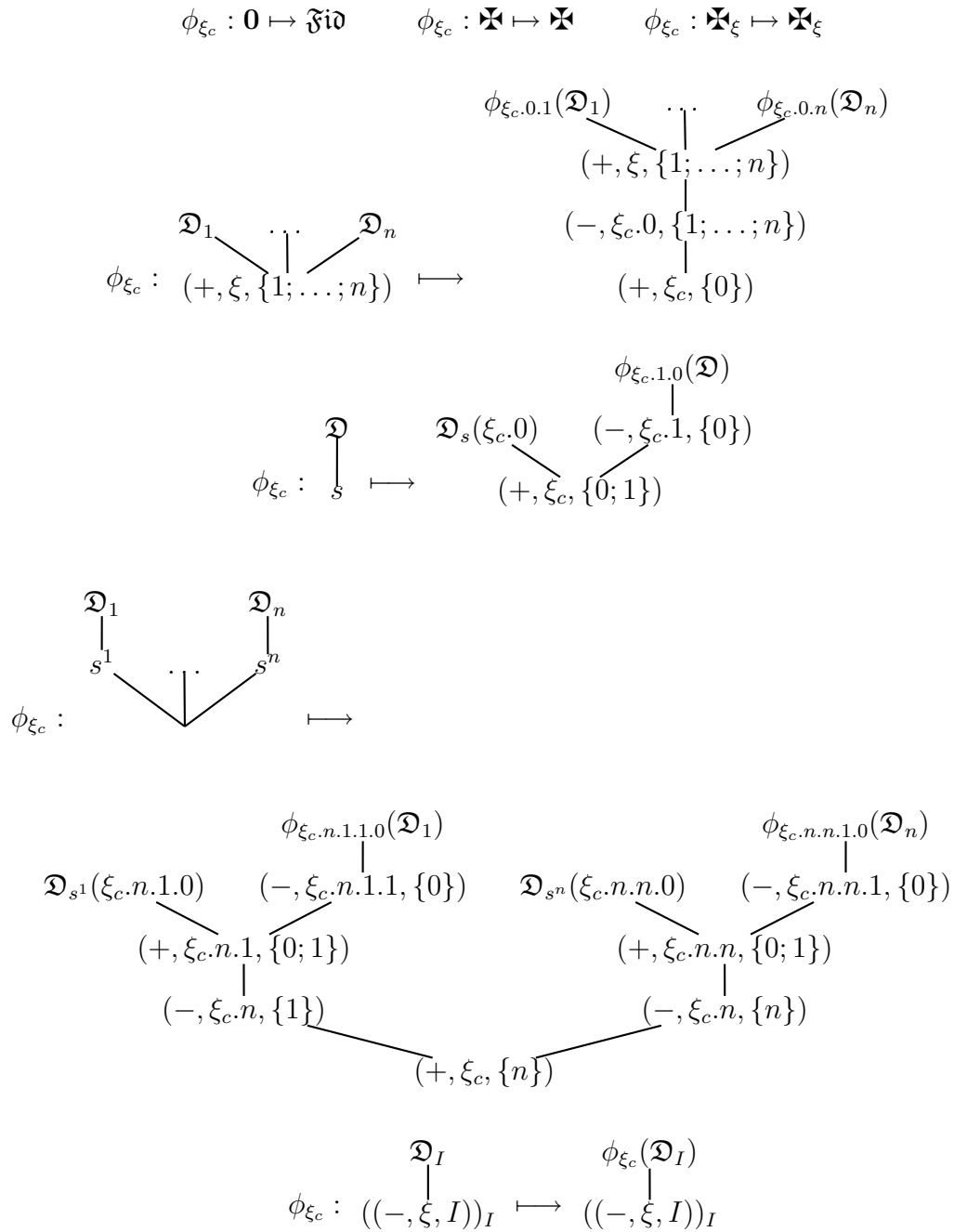


Les actions $(+, \xi, \{0\})$ et $(+, \xi, \{1\})$ focalisent sur le même lieu. Les actions négatives au-dessus de leur ancêtre commun $(+, \xi_c, \{2\})$ doivent donc focaliser aussi sur le même lieu (pour ne pas être dans la même tranche). C'est pourquoi on utilise des actions sur le lieu $\xi_c.2$: $(-, \xi_c.2, \{1\})$ et $(-, \xi_c.2, \{2\})$.

Ces exemples motivent la définition formelle suivante :

Définition 3.27 – Traduction vers la ludique simple

Étant donné un lieu ξ_c , un dessin positivement probabiliste sur une base β disjointe de ξ_c est **traduit** par la fonction récursive ϕ_{ξ_c} de la manière suivante :



Définition 3.28 – Traduction d'un réseau

La **traduction d'un réseau** de desseins positivement probabilistes $\{\mathfrak{D}_1, \dots, \mathfrak{D}_n\}$ consiste à :

- choisir un lieu ξ_c disjoint des lieux des bases des (\mathfrak{D}_i) ;
- et traduire chaque \mathfrak{D}_i avec $\xi_c.i$ comme lieu de traduction des coefficients.

Proposition 3.7 – Traduction d'un réseau

Le traduit d'un réseau est un réseau.

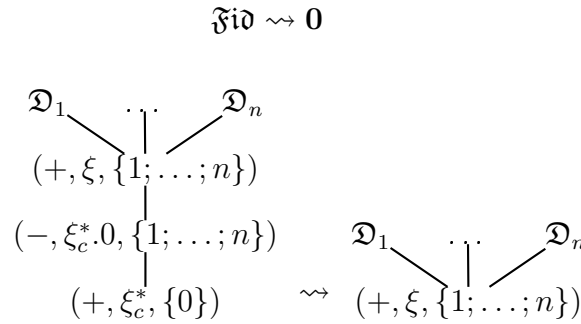
Démonstration : Il suffit de vérifier que la relation de coupure est toujours connexe et acyclique. ■

La traduction inverse n'a pas à se préoccuper de la gestion des sous-lieux de ξ_c comme la traduction devait le faire. Il est donc possible de donner des règles très simples pour inverser la traduction.

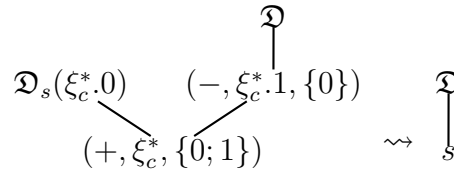
Définition 3.29 – Traduction inverse d'un dessin

La **traduction inverse** d'un dessin simple traduit est l'unique forme normale obtenue par réécriture (infinie) suivant les règles confluentes suivantes (sur les arbres probabilistes) :

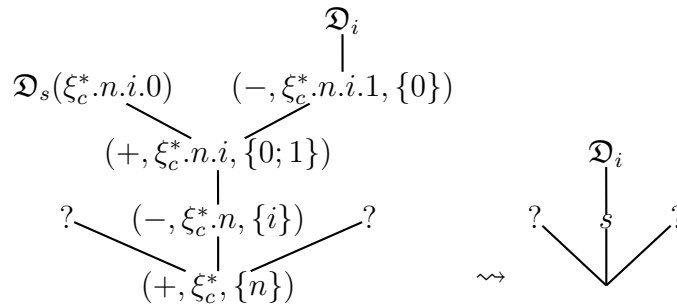
On note ξ_c^* un sous-lieu de ξ_c quelconque.



(et le cas sans continuation)
 $(+, \xi_c^*, \{0\}) \rightsquigarrow \mathbf{0}$



La traduction inverse d'un nœud commençant par $(+, \xi_c^*, \{n\})$ se fait sur toutes les branches en même temps de la manière suivante (les ? indiquent qu'il faut faire la même opération sur les autres branches) :



La justification que la traduction inverse d'un réseau est correcte repose sur la remarque suivante :

Remarque 3.7 Un réseau simple obtenu par traduction d'un réseau de desseins positivement probabilistes n'est pas quelconque : les motifs (morceaux de preuve) sur des lieux ξ_c^* rajoutés par traduction (définition 3.27 page 88) sont tous stables au sens que si l'action du bas sur un lieu ξ_c^* est présente dans la forme normale d'un réseau, alors le motif

en entier est encore présent (à part éventuellement les actions négatives du haut, ce qui représente l'image d'une feuille $\mathbf{0}$).

Définition 3.30 – Normalisation en ludique probabiliste

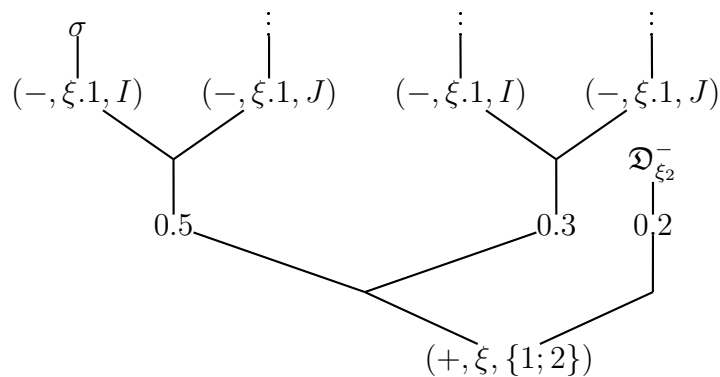
La **normalisation** $\llbracket \mathfrak{R} \rrbracket$ d'un réseau probabiliste \mathfrak{R} de desseins positivement probabilistes s'effectue ainsi

1. Associer un réseau simple \mathfrak{R}_t à ce réseau de desseins \mathfrak{R} par la traduction (définition 3.28 page 89).
2. Normaliser le réseau simple \mathfrak{R}_t en un dessin simple \mathfrak{D}_t .
3. Remettre les scalaires (définition 3.29 page précédente) sur \mathfrak{D}_t . On obtient un dessin $\mathfrak{D} = \llbracket \mathfrak{R} \rrbracket$.

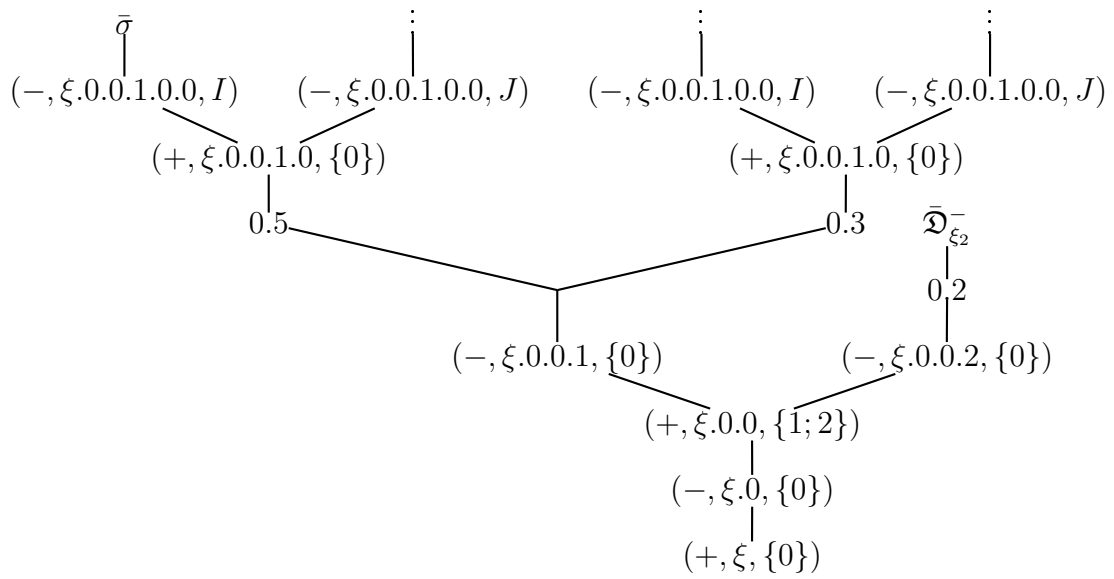
La traduction d'un dessin non positivement probabiliste est plus compliquée car il faut prendre en compte ses coefficients négatifs. On utilise une technique de traduction vers les desseins positivement probabilistes. La traduction change la structure des lieux. Elle peut s'apparenter à une CPS traduction car elle ajoute des interactions pour imposer un ordre d'évaluation.

L'étude d'un exemple est utile.

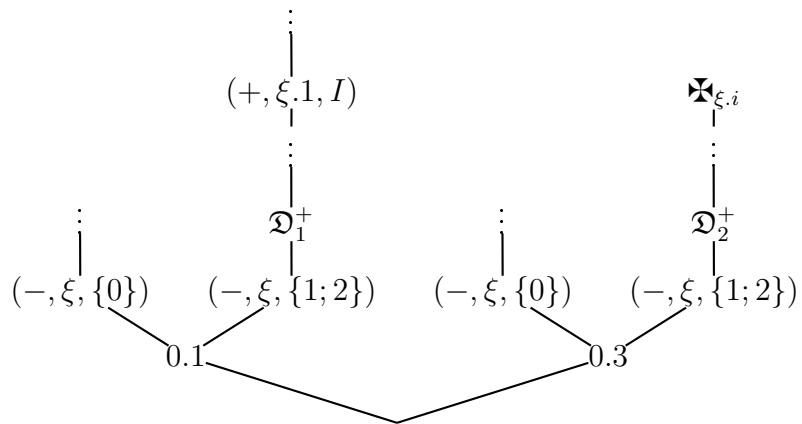
Le dessin :



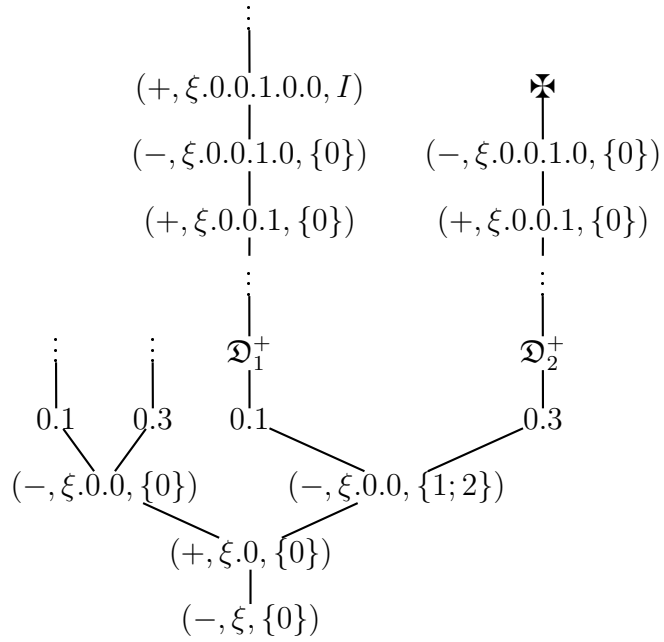
est traduit en



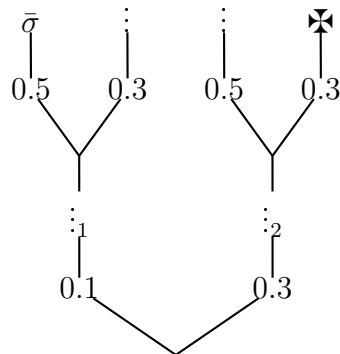
Pour un dessein négatif, il faut faire une transformation similaire :



est traduit en



La normalisation de ces deux desseins donne



Définition 3.31 – Traduction en desseins positivement probabilistes

La traduction consiste à ajouter une action négative avant et une action positive (dupliquée) après chaque coefficient préfixant des desseins négatifs. Elle augmente la longueur des lieux (un lieu ξ est transformé en $\bar{\xi}$).

– une action $(+, \xi, I)$ est traduite en

$$\begin{array}{c} (+, \bar{\xi}.0.0, I) \\ | \\ (-, \bar{\xi}.0, \{0\}) \\ | \\ (+, \bar{\xi}, \{0\}) \end{array}$$

– une action \blackcross_{ξ} est traduite en

$$\begin{array}{c} \blackcross \\ | \\ (-, \bar{\xi}.0, \{0\}) \\ | \\ (+, \bar{\xi}, \{0\}) \end{array}$$

– un dessein négatif $c.(\mathcal{D}_1, \dots, \mathcal{D}_n)$ sur un lieu ξ est traduit en

$$\begin{array}{c} \bar{\mathcal{D}}_1 \qquad \qquad \qquad \bar{\mathcal{D}}_n \\ | \qquad \qquad \qquad | \\ (+, \bar{\xi}.0, \{0\}) \quad \dots \quad (+, \bar{\xi}.0, \{0\}) \\ \diagdown \qquad \qquad \qquad \diagup \\ \bar{c} \\ | \\ (-, \bar{\xi}, \{0\}) \end{array}$$

où \bar{c} est égal à c dans lequel les feuilles nulles ont été remplacées par des actions $(+, \bar{\xi}.0, \emptyset)$ (il y a un petit abus de notation puisque des coefficients et des actions sont mélangés). Cette étape est expliquée à la remarque 3.8.

– un dessein négatif $((-, \xi, I).\mathcal{D}_I)_I$ est traduit en

$$\begin{array}{c} \bar{\mathcal{D}}_I \\ | \\ (-, \bar{\xi}.0.0, I) \end{array})_I$$

Remarque 3.8 La traduction d'un dessein négatif $c.(\mathcal{D}_1, \dots, \mathcal{D}_n)$ sur un lieu ξ enlève les feuilles nulles $\mathbf{0}$ et les remplace par des actions qui ne peuvent pas apparaître à cet endroit autrement (les actions $(+, \bar{\xi}.0, \emptyset)$). Sans feuille nulle, le coefficient \bar{c} n'est plus inclus que dans lui-même (définition 3.11 page 76). Ainsi, la traduction inverse peut se contenter d'agir sur des coefficients \bar{c} (traduits de coefficients négatifs) sans feuille nulle. Sur ces coefficients, elle est donc monotone.

Sans cette condition, l'inclusion sur les positifs

$$\mathbf{0} \subseteq \mathcal{D}^+$$

“polluerait” l'ordre sur les négatifs et donnerait

$$\mathbf{0} \subseteq \mathcal{D}^-$$

ce qui est refusé (car l'incarnation d'un dessein sans $\mathbf{0}$ doit être sans $\mathbf{0}$ pour le théorème de correction).

On étend naturellement la traduction inverse aux desseins probabilistes non positivement probabilistes (en inversant les définitions).

Proposition 3.8 – Monotonie de la traduction inverse

| *La traduction inverse est monotone.*

Démonstration : C'est clair pour les coefficients positifs. Pour les coefficients négatifs, cela provient de la remarque 3.8 page ci-contre. ■

3.3.2 Plongement de la ludique simple

Le plongement canonique de la ludique simple en ludique probabiliste présenté ici sera généralisé à la section 4.2 page 125.

Le résultat important de cette section est le principe des équations génériques (proposition 3.10 page suivante).

Définition 3.32 – Plongement canonique des desseins simples

| *Le **plongement canonique** des desseins simples dans les desseins probabilistes consiste à traduire :*

- $\mathfrak{I}id$ en $\mathbf{0}$
- $\mathfrak{I}id^-$ en $\mathbf{0}$
- les autres nœuds σ en le coefficient $\mathbf{1}$ suivi de σ .

On remarque que le traduit en ludique simple d'un dessein simple \mathfrak{D} plongé en ludique probabiliste est \mathfrak{D} lui-même. Cela donne la proposition suivante :

Proposition 3.9 – Plongement des desseins simples

| *Le plongement canonique des desseins simples commute avec la normalisation.*

Démonstration : La normalisation par codage (section 3.3.1 page 82) permet de conclure. ■

Les liens étroits entre normalisation en ludique simple et ludique probabiliste sont éclairés par la notion d'équation générique.

Définition 3.33 – Équations génériques

| *Les **équations génériques** sont définies par la grammaire :*

$$E ::= \mathfrak{R} = \mathfrak{R}$$

$$\mathfrak{R} ::= \mathfrak{X} \mid \mathfrak{X}_\xi \mid \mathfrak{I}id \mid \mathfrak{D}_i \mid \mathfrak{D}_i, \mathfrak{R} \mid \llbracket \mathfrak{R} \rrbracket \mid \mathfrak{R}_j \mid \mathfrak{R}_j, \mathfrak{R}$$

| où les $\mathfrak{D}_i, \mathfrak{R}_i$ sont des terminaux.

Définition 3.34 – Évaluation d’une équation générique

Une **équation générique** $E(\mathfrak{D}_{i_1}, \dots, \mathfrak{D}_{i_n}, \mathfrak{R}_{j_1}, \dots, \mathfrak{R}_{j_m})$ est vraie sur une famille \mathcal{F} de desseins (simples ou probabilistes) si pour tous desseins $(\mathfrak{D}_{i_1}, \dots, \mathfrak{D}_{i_n})$ de \mathcal{F} et tous réseaux $\mathfrak{R}_{j_1}, \dots, \mathfrak{R}_{j_m}$ composés de desseins de \mathcal{F} tels que les deux membres de l’équation E sont bien formés alors E est vérifiée.

Proposition 3.10 – Principe des équations génériques

Toute équation générique E est vraie sur les desseins simples si et seulement si elle est vraie sur les desseins probabilistes.

Démonstration : C’est une conséquence de la normalisation par codage : par définition, la normalisation commute à la traduction.

- ⇐ Soit E vraie sur les desseins probabilistes. Comme les desseins simples se plongent dans les desseins probabilistes, E est vraie sur les desseins simples.
- ⇒ Soit E vraie sur les desseins simples. Soient $\mathfrak{D}_{i_1}, \dots, \mathfrak{D}_{i_n}$ des desseins probabilistes et $\mathfrak{R}_{j_1}, \dots, \mathfrak{R}_{j_m}$ des réseaux probabilistes tels que les deux membres de l’équation E sont bien formés. Soient $\tilde{\mathfrak{D}}_{i_1}, \dots, \tilde{\mathfrak{D}}_{i_n}$ et $\tilde{\mathfrak{R}}_{j_1}, \dots, \tilde{\mathfrak{R}}_{j_m}$ leurs traductions dans les desseins simples. Les desseins $\tilde{\mathfrak{D}}_{i_1}, \dots, \tilde{\mathfrak{D}}_{i_n}$ et les réseaux $\tilde{\mathfrak{R}}_{j_1}, \dots, \tilde{\mathfrak{R}}_{j_m}$ sont tels que $E(\tilde{\mathfrak{D}}_i, \tilde{\mathfrak{R}}_j)$ est bien formée d’après la proposition 3.7 page 89. Par hypothèse, $E(\tilde{\mathfrak{D}}_i, \tilde{\mathfrak{R}}_j)$ est vraie. Donc, par injectivité de la traduction, $E(\mathfrak{D}_i, \mathfrak{R}_j)$ est vraie donc E est vérifiée sur les desseins probabilistes. ■

Cette proposition sera utilisée par exemple pour montrer la propriété d’associativité de la ludique probabiliste (proposition 3.26 page 110).

Définition 3.35 – Orthogonalité

Deux desseins \mathfrak{D} et \mathfrak{E} de bases opposées sont **orthogonaux** si

$$\mathcal{W}(\llbracket \mathfrak{D}, \mathfrak{E} \rrbracket) = 1$$

Remarque 3.9 Cette définition n’est pas forcément la seule possible. On pourrait prendre comme condition d’orthogonalité par exemple

$$\mathcal{W}(\llbracket \mathfrak{D}, \mathfrak{E} \rrbracket) > 0$$

ou

$$\mathcal{W}(\llbracket \mathfrak{D}, \mathfrak{E} \rrbracket) \geq \alpha$$

pour un certain $\alpha \in]0; 1[$.

On pourrait aussi prendre comme condition d’orthogonalité des choses plus baroques comme

$$\mathcal{W}(\llbracket \mathfrak{D}, \mathfrak{E} \rrbracket) \text{ est rationnel}$$

ou même

$$\llbracket \mathfrak{D}, \mathfrak{E} \rrbracket \text{ est un arbre symétrique}$$

Il n’est pas clair que cela donne des choses utilisables...

3.3.3 Normalisation par tranches

On peut également définir la normalisation directement en ludique probabiliste. L'idée est de normaliser un réseau en oubliant la structure additive et probabiliste puis en recollant les morceaux obtenus. Les morceaux pratiques pour faire cette opération (découpe, normalisation puis recollage) sont les c -tranches : des tranches dans lesquelles on a gardé suffisamment d'information de coefficient pour recoller.

L'intuition calculatoire est que tout fil de calcul lors de l'interaction entre deux desseins décrit une tranche. C'est vrai pour la ludique simple [Fag02] et pour la version linéaire de la ludique probabiliste présentée dans ce chapitre. Cela n'est plus vrai en présence de répétitions (partie III page 171).

Définition 3.36 – c -tranche

Une c -tranche est un dessin tel que :

- chaque coefficient positif a au plus une feuille vide (i.e. est suivi d'au plus une action positive immédiate) ;
- et chaque coefficient négatif est suivi par au plus une action négative immédiate.

Remarque 3.10 Un cas particulier de c -tranches est la c -tranche constituée d'un unique coefficient sans feuille vide.

Définition 3.37 – c -tranche d'un dessin

Une c -tranche \mathfrak{S} est une c -tranche d'un dessin \mathfrak{D} si \mathfrak{S} est incluse dans \mathfrak{D} .

Comme pour les tranches, on utilise aussi le terme de c -tranche pour un réseau formé de c -tranches.

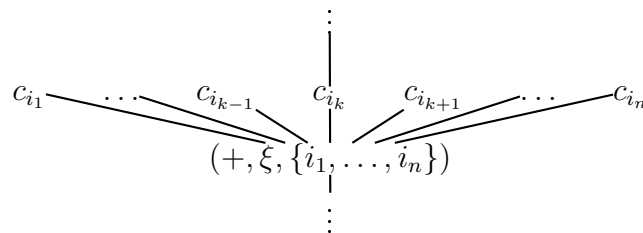
La définition suivante est celle qui correspond aux chroniques dans le cadre des c -tranches. L'intérêt est que l'ensemble des c -chroniques d'un dessin détermine ce dessin.

Définition 3.38 – c -chronique

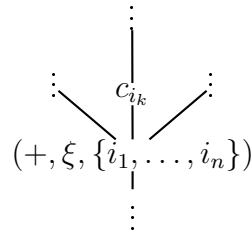
Une c -chronique est une c -tranche telle que

- chaque action positive $(+, \xi, \{i_1, \dots, i_n\})$ avec $n \geq 1$ est suivie par au plus un coefficient suivi par au moins une action négative immédiate.
- chaque action négative est suivie par un coefficient différent de 1 ou par au moins une action positive.

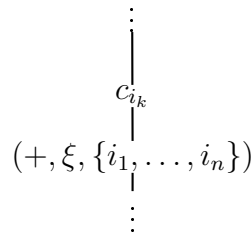
Une chronique est de la forme



est simplement notée :



ou même, s'il n'y a pas d'ambiguïté :



Remarque 3.11 *Un cas particulier de c -chronique est la c -chronique constituée d'un unique coefficient c .*

La condition sur les actions négatives correspond au cas de la ludique simple qui stipule que les actions maximales sont positives ou égales à \mathfrak{Fid}^- .

Proposition 3.11 – c -chroniques

| *Tout dessein est en bijection avec l'ensemble de ses c -chroniques.*

Démonstration : Cela provient de la définition même de dessein (définition 3.13 page 77).

Si deux desseins ont les mêmes ensembles de c -chroniques, on montre par récurrence sur $n \in \mathbb{N}$ qu'ils sont égaux jusqu'à la hauteur n et donc sur les hauteurs finies. On en déduit qu'ils sont égaux. ■

Définition 3.39 – Cohérence de c -chroniques

| *Deux c -chroniques sont **cohérentes** si elles sont incluses dans un même dessein.*

Définition 3.40 – c -préfixe

| *Une c -chronique \mathfrak{c} est un **c -préfixe** d'une c -chronique \mathfrak{c}' si \mathfrak{c} est incluse dans \mathfrak{c}' .*

Proposition 3.12 – Caractérisation des desseins par les c -chroniques

| *Un ensemble de c -chroniques sur une base β positive (resp. négative) clos par c -préfixe est l'ensemble des c -chroniques d'un dessein si et seulement si il est cohérent.*

Démonstration : On prouve par induction sur la longueur des c -chroniques que le sous-ensemble constitué des c -chroniques de longueur au plus n est un dessein si et seulement si il est cohérent. Le cas général est déduit par approximation. ■

Un réseau constitué de deux c -tranches se normalise de manière similaire à la ludique simple : les deux ordres entre les actions des deux c -tranches sont mélangés et le résultat de la normalisation est l'ordre induit (avec les coefficients) entre les actions restantes.

Dans une c -tranche, chaque coefficient a au plus une feuille vide et est suivi d'au plus une action. On note alors les desseins positifs $c.K^+(\mathcal{D}_1^- \dots \mathcal{D}_n^-)$ et les desseins négatifs $c.(K^-. \mathcal{D}, \dots)$ (pour désigner un dessin négatif de coefficient préfixe c contenant l'action K^- suivie de \mathcal{D}).

Définition 3.41 – Normalisation de c -tranches

La **réduction pas à pas** d'un réseau clos de c -tranches \mathfrak{R} est étiquetée par des coefficients avec au plus une feuille vide. Elle est définie par :

$$\begin{array}{ll}
 c.K^+, \mathfrak{R} & \xrightarrow{c} K^+, \mathfrak{R} \\
 \mathfrak{X}_\xi, c'.(\mathfrak{S}\mathfrak{k}, \dots, \mathfrak{S}\mathfrak{k}, \mathcal{D}_\xi^-, \mathfrak{S}\mathfrak{k}, \dots, \mathfrak{S}\mathfrak{k}), \mathfrak{R} & \xrightarrow{c'} \mathfrak{X} \\
 \mathfrak{X}, \mathfrak{R} & \xrightarrow{1} \mathfrak{X} \\
 K^+(\mathcal{D}_1^- \dots \mathcal{D}_n^-), c'.(\mathfrak{S}\mathfrak{k}, \dots, \mathfrak{S}\mathfrak{k}, \mathcal{D}_\xi^-, \mathfrak{S}\mathfrak{k}, \dots, \mathfrak{S}\mathfrak{k}), \mathfrak{R} & \xrightarrow{c'} K^+(\mathcal{D}_1^- \dots \mathcal{D}_n^-), \mathcal{D}_\xi^-, \mathfrak{R} \\
 K^+(\mathcal{D}_1^- \dots \mathcal{D}_n^-), (K^-. \mathcal{D}^+ \dots), \mathfrak{R} & \xrightarrow{1} \mathcal{D}^+, \mathcal{D}_1^-, \dots, \mathcal{D}_n^-, \mathfrak{R} \\
 K^+(\mathcal{D}_1^- \dots \mathcal{D}_n^-), \mathfrak{R} & \xrightarrow{1} \mathbf{0} \text{ si } K^- \notin \mathfrak{R}
 \end{array}$$

où

- $\mathfrak{S}\mathfrak{k}$ est une abréviation pour le sconse $\mathfrak{S}\mathfrak{k}\mathfrak{u}\mathfrak{n}\mathfrak{k}$;
- \mathcal{D}_ξ^- est un dessin semi-simple ;
- c' est le coefficient c' dans lequel les feuilles vides contenant $\mathfrak{S}\mathfrak{k}$ sont remplacées par des feuilles nulles.

La **normalisation** $\llbracket \mathfrak{R} \rrbracket_t$ d'un réseau clos \mathfrak{R} consiste à recoller les coefficients étiquetant la réduction pas à pas. Elle est définie par :

$$\llbracket \mathfrak{R} \rrbracket_t = \begin{cases} c_1 \dots c_n \cdot \mathfrak{X} & \text{si } \mathfrak{R} \xrightarrow{c_1} \dots \xrightarrow{c_n} \mathfrak{X} \\
 c_1 \dots c_n \cdot \mathbf{0} & \text{si } \mathfrak{R} \xrightarrow{c_1} \dots \xrightarrow{c_n} \mathbf{0} \\
 c_1 \dots c_n \dots \mathbf{0} & \text{si } \mathfrak{R} \xrightarrow{c_1} \xrightarrow{c_2} \dots \xrightarrow{c_n} \dots
 \end{cases}$$

Le cas $\mathfrak{R} \xrightarrow{c_1} \xrightarrow{c_2} \dots \xrightarrow{c_n} \dots$ signifie que \mathfrak{R} est réduit une infinité d'étapes sans tomber sur le démon ou sur $\mathbf{0}$. Le coefficient $c_1 \dots c_n \dots \mathbf{0}$ est le recollement (potentiellement infini) des coefficients c_i . Si ce coefficient est fini (ce qui arrive si l'ensemble des i tel que c_i est différent de $\mathbf{1}$ est fini) alors la (dernière) feuille vide est remplacée par $\mathbf{0}$.

Remarque 3.12 Les coefficients c des étiquettes dans la définition précédente peuvent n'avoir aucune feuille vide. Dans ce cas, le recollement de c et c' est égal à c .

Pour la suite de la section, on note $c.\mathfrak{R}$ le réseau constitué de desseins de \mathfrak{R} tel que c préfixe le dessin principal de \mathfrak{R} .

Proposition 3.13 – Quelques propriétés de la traduction en ludique simple

Soit $Trad$ la traduction (définition 3.28 page 89) de la ludique probabiliste en ludique simple.

– Soit c un coefficient à au plus une feuille vide et \mathfrak{D} un dessein.

On a

$$Trad(c.Trad(\mathfrak{D})) = Trad(c.\mathfrak{D})$$

– On note $\llbracket \cdot \rrbracket_s$ la normalisation en ludique simple.

Soit c un coefficient à au plus une feuille vide et \mathfrak{R} un réseau.

$$\llbracket Trad(c.\mathfrak{R}) \rrbracket_s = Trad(c. \llbracket Trad(\mathfrak{R}) \rrbracket_s)$$

Démonstration : On ne détaille pas la gestion du lieu ξ_c rajouté dans la traduction.

– C’est immédiat.

– La traduction d’un dessein $c.\mathfrak{D}$ en ludique simple fait intervenir un arbre sur le lieu ξ_c puis la traduction de \mathfrak{D} . L’arbre sur le lieu ξ_c est “conservé” par normalisation. ■

Proposition 3.14 – Traduction en ludique simple et $\mathfrak{R} \xrightarrow{c} \mathfrak{R}'$

En prenant les notations de la définition 3.41 page précédente, si $\mathfrak{R} \xrightarrow{c} \mathfrak{R}'$ alors

$$\llbracket Trad(\mathfrak{R}) \rrbracket_s = \llbracket Trad(c.\mathfrak{R}') \rrbracket_s$$

Démonstration : La preuve se fait par cas suivant la définition 3.41 page précédente.

Tous les cas sont immédiats. ■

Proposition 3.15 – Validité de la définition de normalisation des c -tranches

Sur les c -tranches, les fonctions $\llbracket \cdot \rrbracket$ et $\llbracket \cdot \rrbracket_t$ coïncident.

Démonstration : On a

$$\llbracket \mathfrak{R} \rrbracket_t = \begin{cases} c_1 \dots c_n \cdot \mathfrak{X} & \text{si } \mathfrak{R} \xrightarrow{c_1} \dots \xrightarrow{c_n} \mathfrak{X} \\ c_1 \dots c_n \cdot \mathbf{0} & \text{si } \mathfrak{R} \xrightarrow{c_1} \dots \xrightarrow{c_n} \mathbf{0} \\ c_1 \dots c_n \dots \mathbf{0} & \text{si } \mathfrak{R} \xrightarrow{c_1} \xrightarrow{c_2} \dots \xrightarrow{c_n} \dots \end{cases}$$

– Cas 1. $\mathfrak{R} \xrightarrow{c_1} \mathfrak{R}_1 \dots \xrightarrow{c_n} \mathfrak{X}$.

On a $\llbracket \mathfrak{R} \rrbracket_t = c_1 \dots c_n \cdot \mathfrak{X}$

On procède par récurrence sur n .

$$\begin{aligned} \llbracket Trad(\mathfrak{R}) \rrbracket_s &= \llbracket Trad(c_1.\mathfrak{R}_1) \rrbracket_s && \text{par la proposition 3.14} \\ &= Trad(c_1. \llbracket \mathfrak{R}_1 \rrbracket_s) && \text{par la proposition 3.13} \\ &= \dots \\ &= Trad(c_1.c_2 \dots \llbracket \mathfrak{X} \rrbracket_s) && \text{par récurrence} \\ &= Trad(c_1.c_2 \dots \mathfrak{X}) \end{aligned}$$

Donc

$$\llbracket \mathfrak{R} \rrbracket = \llbracket \mathfrak{R} \rrbracket_t$$

- Cas 2. $\mathfrak{R} \xrightarrow{c_1} \dots \xrightarrow{c_n} \mathbf{0}$.
On procède comme pour le cas 1.
- Cas 3. $\mathfrak{R} \xrightarrow{c_1} \xrightarrow{c_2} \dots \xrightarrow{c_n} \dots$
On procède comme pour le cas 1. ■

Définition 3.42 – Cohérence des c -tranches

Deux c -tranches sont **cohérentes** si elles sont c -tranches d'un même dessein ou réseau.

Proposition 3.16 – Cohérence des résultats des normalisations

Les résultats de la normalisation de deux c -tranches d'un réseau \mathfrak{R} sont des c -tranches cohérentes.

Démonstration : On peut le vérifier par codage en ludique simple : on traduit la définition de c -tranche dans les traduits des desseins probabilistes. La proposition découle du fait que leurs normalisations sont cohérentes car ce sont des sous-desseins simples de la normalisation du traduit.

Une vérification directe est plus laborieuse. ■

Définition 3.43 – Union de c -tranches

L'**union** d'un ensemble de c -tranches est, s'il existe, le plus petit dessein qui les contient.

L'union de c -tranches permet de définir la normalisation :

Proposition 3.17 – Normalisation d'un réseau

$$[[\mathfrak{R}]] = \bigcup_{\mathfrak{G} \subseteq \mathfrak{R}, \mathfrak{G} \text{ une } c\text{-tranche}} [[\mathfrak{G}]]$$

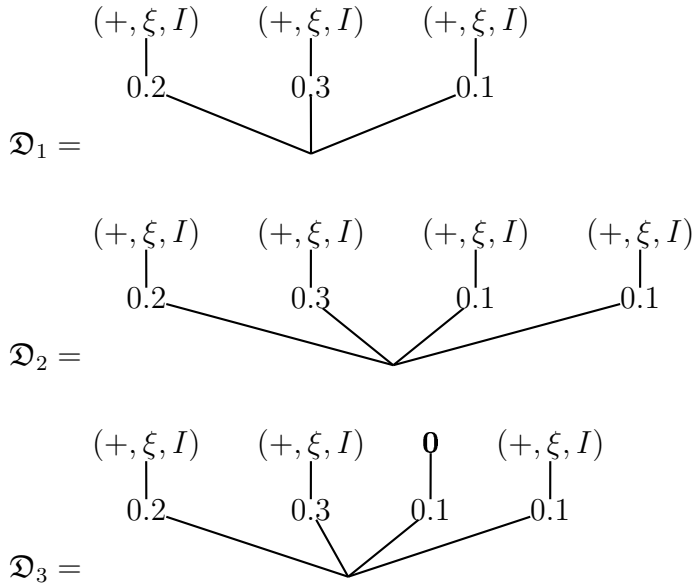
3.4 Ordres

Dans cette section, on définit les versions probabilistes des ordres stable et extensionnel (voir section 1.2.3 page 30). Ils interviennent dans les théorèmes de stabilité 3.25 page 110 et de séparation 3.21 page 108. L'ordre stable est relié à la notion d'inclusion. Ici, la notion utile est celle d'*inclusion pleine* :

Définition 3.44 – Inclusion pleine

Un dessein \mathfrak{D} est **pleinement inclus** dans un autre dessein \mathfrak{D}' si \mathfrak{D} est obtenu à partir de \mathfrak{D}' en remplaçant des coefficients positifs par la feuille nulle (en oubliant leur continuation) et en enlevant des actions négatives. On note cette situation $\mathfrak{D} \subseteq \mathfrak{D}'$.

L'idée derrière cette définition est que l'inclusion pleine permet de *suivre* le plongement d'un dessein dans un autre. Par exemple, définissons trois desseins



Le dessin \mathfrak{D}_1 n'est pas pleinement inclus dans \mathfrak{D}_2 alors que \mathfrak{D}_3 l'est. Le dessin \mathfrak{D}_1 n'est pas pleinement inclus dans \mathfrak{D}_2 car son coefficient de base n'a que trois feuilles alors que celui de \mathfrak{D}_2 en a quatre. Ainsi, typiquement, on ne sait pas à quelle feuille dans \mathfrak{D}_2 correspond la feuille de droite dans \mathfrak{D}_1 . Par contre, le dessin \mathfrak{D}_3 a une géométrie suffisamment proche de celle de \mathfrak{D}_2 pour associer les deux feuilles les plus à droite.

Comme pour la ludique simple, l'ordre stable en ludique probabiliste est l'ordre d'inclusion.

Définition 3.45 – Ordre stable

L'ordre stable est l'ordre engendré par :

- Cas positif.

$$\mathbf{0} \sqsubseteq \mathfrak{D}^+$$
- Cas négatif si $N \subseteq M$ alors

$$((-, \xi, I) \cdot \mathfrak{D}_I)_{I \in N} \sqsubseteq ((-, \xi, I) \cdot \mathfrak{D}_I)_{I \in M}$$

Remarque 3.13 On n'utilise pas la règle suivante :

$$\text{si pour tout } i, s_i \leq s'_i \text{ alors } (s_1 \cdot \mathfrak{D}_1, \dots, s_n \cdot \mathfrak{D}_n) \sqsubseteq (s'_1 \cdot \mathfrak{D}_1, \dots, s'_n \cdot \mathfrak{D}_n)$$

De même qu'en ludique simple, la divergence négative n'est pas stablement inférieure aux autres desseins négatifs. Ainsi, on n'utilise pas la règle suivante (avec $\mathfrak{Fid}^- = \mathbf{0}$) :

$$\mathfrak{Fid}^- \sqsubseteq \mathfrak{D}^-$$

L'ordre extensionnel \preceq est le plus petit ordre entre desseins contenant l'ordre stable, le remplacement des actions par des démons (focalisés ou non) ainsi que le remplacement de la divergence négative par un autre dessin négatif. Naturellement, un démon focalisé ne remplace qu'une action positive *sur le même lieu*.

Définition 3.46 – Ordre extensionnel

L'ordre extensionnel est l'ordre engendré par :

– Cas positif.

$$\mathbf{0} \preceq \mathfrak{D}^+ \quad \mathfrak{D}^+ \preceq \mathfrak{K} \quad (+, \xi, I). \mathfrak{K} \preceq \mathfrak{K}_\xi$$

– Cas négatif.

$$\mathbf{0} \preceq \mathfrak{D}^-$$

et, si $N \subseteq M$ alors

$$((-, \xi, I). \mathfrak{D}_I)_{I \in N} \preceq ((-, \xi, I). \mathfrak{D}_I)_{I \in M}$$

3.5 Théorèmes analytiques

Les théorèmes analytiques s'étendent au cadre probabiliste. La preuve du théorème de séparation est reliée à la géométrie des coefficients, celle du théorème de stabilité au codage et l'associativité est une conséquence directe du principe des équations génériques (proposition 3.10 page 96).

3.5.1 Séparation

Définition 3.47 – Ensemble des scalaires d'un dessin

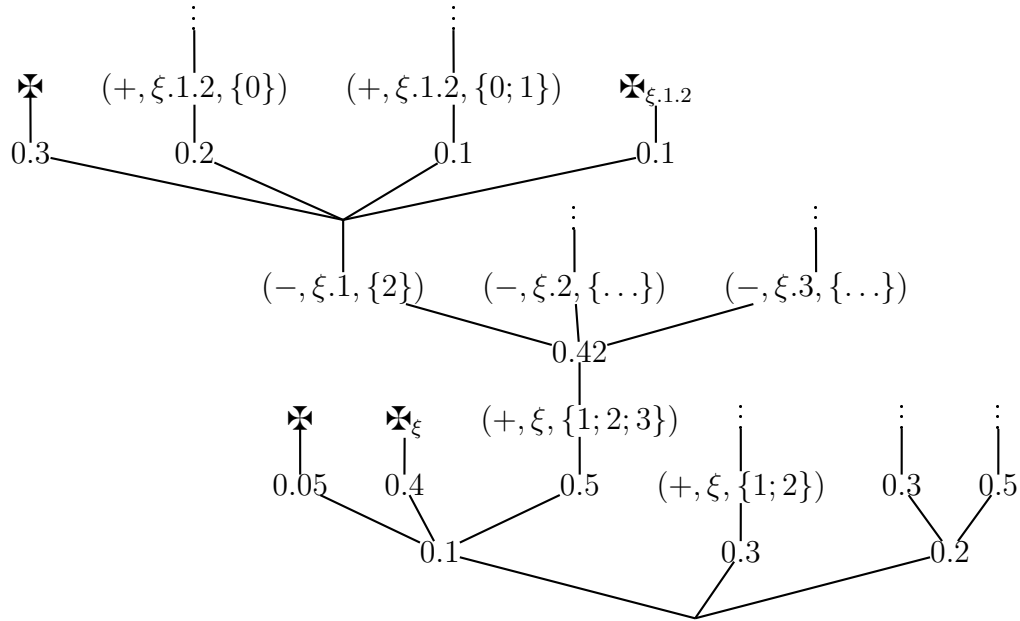
L'ensemble des scalaires $\text{Scal}(\mathfrak{D})$ d'un dessin \mathfrak{D} est l'ensemble composé des scalaires apparaissant dans les coefficients de \mathfrak{D} .

L'ensemble des scalaires généralisés $\text{Scal}^(\mathfrak{D})$ d'un dessin \mathfrak{D} est l'ensemble composé des scalaires apparaissant dans les coefficients de \mathfrak{D} et des puissances négatives de deux.*

$$\text{Scal}^*(\mathfrak{D}) = \text{Scal}(\mathfrak{D}) \cup \{2^{-n} \mid n \in \mathbb{N}\}$$

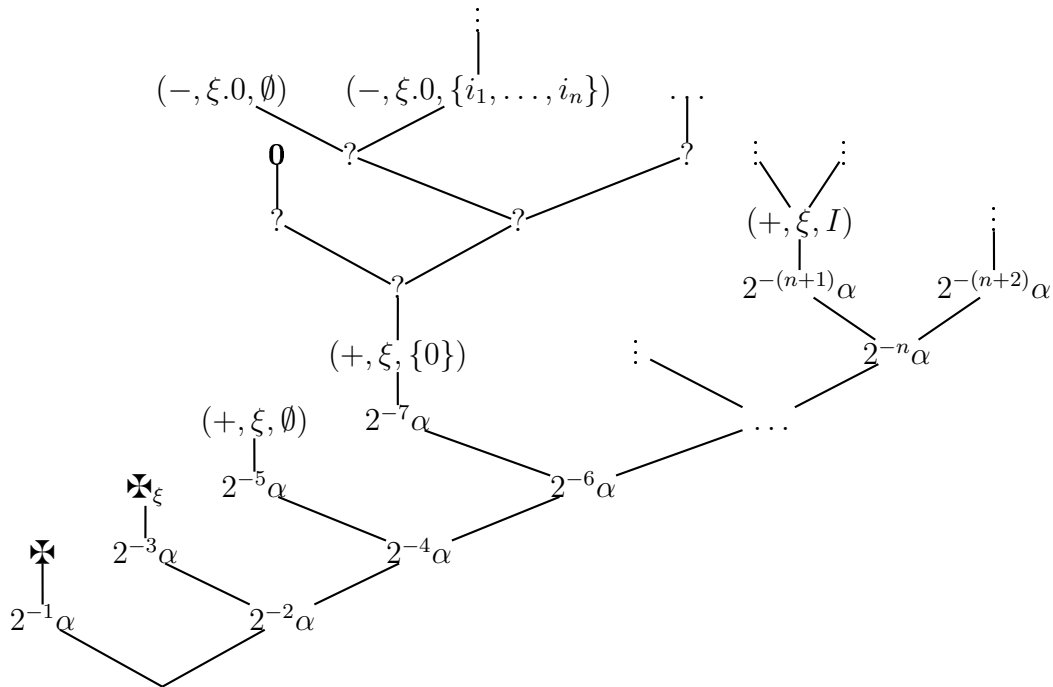
Remarque 3.14 *Les puissances négatives de 2 apparaissent dans la définition précédente car l'ensemble $\text{Scal}^*(\mathfrak{D})$ sera calculé (proposition 3.18 page 107) par interaction avec un dessin qui amènera naturellement des puissances négatives de 2.*

La notion fondamentale pour la preuve du théorème de séparation en ludique probabiliste est la notion de dessin universel : un dessin qui *teste tout ce qu'il peut*. Un premier exemple est utile. Sur la base $\vdash \xi$, les dessins sont de la forme :



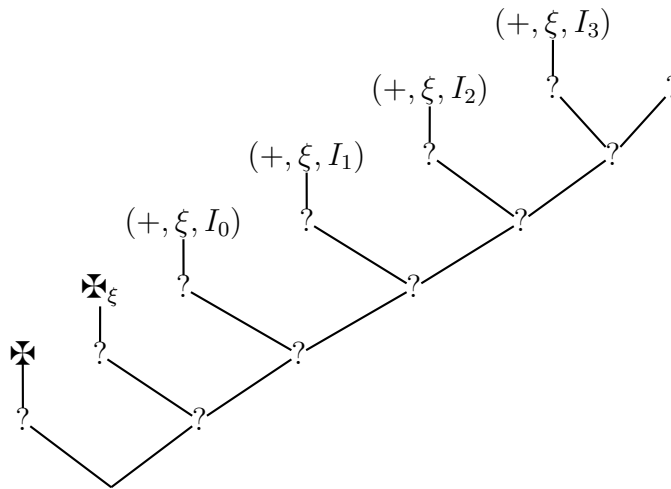
C'est-à-dire qu'au premier étage, il y a un coefficient avec des feuilles vides puis des actions $(+, \xi, I)$, des démons non focalisés, et des démons focalisés \blackboxtimes_{ξ} . À l'étage suivant, il y a des coefficients, des actions négatives $(-, \xi.i, J)$ et ça recommence. Ce schéma est suffisamment général pour définir un dessin ayant toutes les chroniques possibles (aux coefficients près) : les desseins universels.

Les desseins universels positifs sont de la forme suivante (aux coefficients près) :

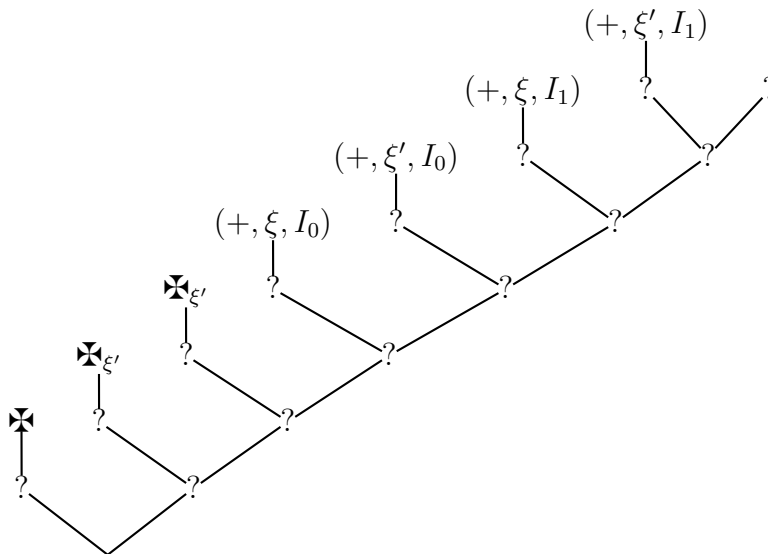


où $\alpha \in]0; 1]$ de telle manière que tous les scalaires qui apparaissent dans le dessin sont distincts et toutes les chroniques sur la base $\vdash \xi$ sont présentes (en oubliant les scalaires).

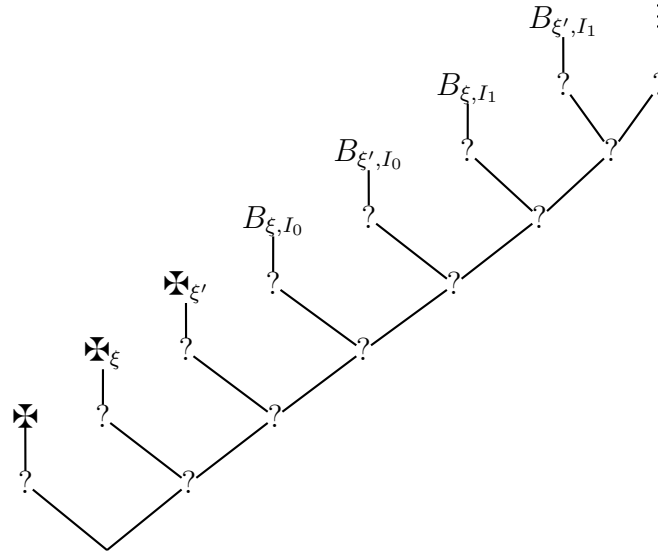
Plus précisément, en notant $?$ un scalaire pour le moment non défini et par (I_n) une énumération injective de $\mathcal{P}_{fin}(\mathbb{N})$, le premier nœud positif est



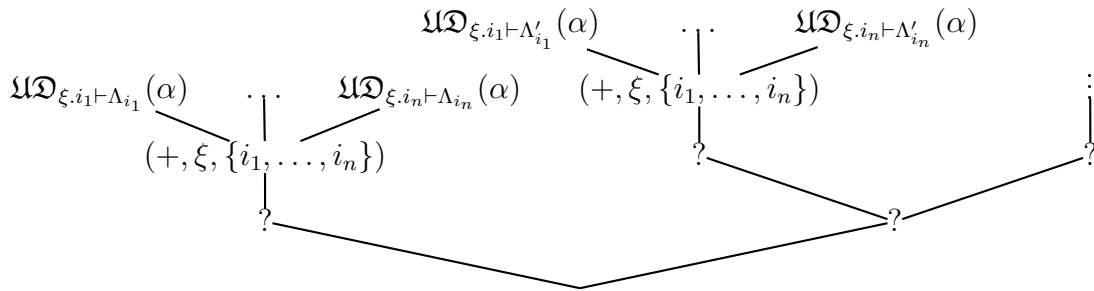
Sur une base multiple, par exemple $\vdash \xi, \xi'$, l'idée est d'alterner les actions $(+, \xi, I)$ et $(+, \xi', I)$ de la manière suivante :



Cela ne suffit pas tout à fait, il faut aussi alterner les partages de contexte. Il faut donc prendre :

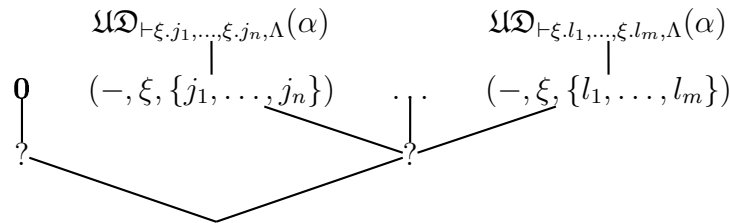


où $B_{\xi, \{i_1, \dots, i_n\}}$ est le dessein :



où les (Λ_{i_k}) et (Λ'_{i_k}) décrivent les partitions de Λ en n parties si la base est $\vdash \xi, \Lambda$.

Le dessein universel sur la base $\xi \vdash \Lambda$ est le dessein



Il reste à décrire les scalaires qui apparaissent dans le dessein universel. Une union dénombrable d'ensembles dénombrables étant dénombrable, les positions pour les scalaires ? sont dénombrables. On se donne une énumération bijective $p = (p_i)_{i \in \mathbb{N}}$ de ces positions. On définit le scalaire ? comme $2^{-(i+1)}\alpha$ où i est l'indice dans p de la position du scalaire ?.

Définition 3.48 – Dessein universel

| Soit β une base. Le **dessein universel** $\mathcal{U}_{\mathcal{D}_{\beta}}(\alpha)$ est le dessein que l'on vient de décrire.

Remarque 3.15 *Les desseins universels sont bien des desseins. De plus, les scalaires d'un dessin universel sont tous différents et leur somme totale est bornée par 1. Ainsi, toutes les sommes partielles de scalaires sont bornées par 1. L'adjectif universel provient de l'observation que tous les desseins simples de base β sont inclus dans le dessin universel de base β au sens des projections simples (définition 3.12 page 77) : l'ensemble des sous-desseins des projections simples du dessin universel de base β est l'ensemble des desseins simples de base β .*

Proposition 3.18 – Scalaires dans $\llbracket \mathfrak{D}, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha) \rrbracket$

Soit \mathfrak{D} un dessin de base unaire β . Pour tout α , on a

$$\text{Scal}(\mathfrak{D}) \subseteq \text{Scal}(\llbracket \mathfrak{D}, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha) \rrbracket) \subseteq \text{Scal}(\mathfrak{D}) \cup \{2^{-n}\alpha \mid n \in \mathbb{N}\}$$

Démonstration : La preuve repose sur plusieurs remarques :

- Après une action négative, le dessin universel $\mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha)$ utilise (entre autres actions) un démon et un démon focalisé sur tous les lieux (positifs) de la base (locale).
- Après une action positive, le dessin universel $\mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha)$ essaie de focaliser sur toutes les actions possibles.
- Les actions positives sont suivies de toutes les actions négatives possibles.

Ce qui implique (par récurrence) que tout nœud de \mathfrak{D} est visité lors de la normalisation du réseau $\mathfrak{D}, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha)$ et que le coefficient associé apparaît dans le résultat $\llbracket \mathfrak{D}, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha) \rrbracket$. ■

Remarque 3.16 *La proposition précédente implique que*

$$\text{Scal}^*(\mathfrak{D}) = \text{Scal}^*(\llbracket \mathfrak{D}, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(1) \rrbracket)$$

Proposition 3.19 – Existence de $\alpha_{\text{Scal}^*(\mathfrak{D})}$

Soit \mathfrak{D} un dessin de base unaire β . Il existe un scalaire $\alpha_{\text{Scal}^*(\mathfrak{D})} \in]0; 1[$ tel que

$$\text{Scal}^*(\mathfrak{D}) \cap \alpha_{\text{Scal}^*(\mathfrak{D})}\mathbb{Q} = \emptyset$$

Démonstration : L'ensemble des scalaires $\text{Scal}^*(\mathfrak{D})$ de \mathfrak{D} est au plus dénombrable donc $\text{Scal}^*(\mathfrak{D}) * \mathbb{Q}$ est au plus dénombrable or $]0, 1[$ n'est pas dénombrable donc il existe un scalaire $\alpha_{\text{Scal}^*(\mathfrak{D})} \in]0; 1[$ tel que $\text{Scal}^*(\mathfrak{D}) \cap \alpha_{\text{Scal}^*(\mathfrak{D})}\mathbb{Q} = \emptyset$. ■

Définition 3.49 – Le scalaire $\alpha_{\mathfrak{D}}$

Soit \mathfrak{D} un dessin de base unaire. Soit $\alpha_{\mathfrak{D}}$ un coefficient $\alpha_{\text{Scal}^*(\mathfrak{D})}$ vérifiant la propriété de la proposition 3.19 ne dépendant que de $\text{Scal}^*(\mathfrak{D})$.

Définition 3.50 – Opposant universel

Soit \mathfrak{D} un dessin sur une base unaire β . L'**opposant universel** de \mathfrak{D} est

$$\mathfrak{U}\mathfrak{D}_{\mathfrak{D}} = \mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha_{\llbracket \mathfrak{D}, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(1) \rrbracket})$$

Remarque 3.17 *Cette définition d'opposant universel est constructive dans le sens qu'elle dépend uniquement des résultats des normalisations de \mathfrak{D} avec d'autres desseins et non de l'introspection de ses scalaires. Les résultats des normalisations de \mathfrak{D} sont observables puisque ce sont essentiellement des coefficients.*

La proposition clef pour la séparation est la suivante :

Proposition 3.20 – Injectivité de $\mathfrak{D} \mapsto \llbracket \mathfrak{D}, \mathfrak{U}\mathfrak{D}_{\mathfrak{D}} \rrbracket$

| Sur une base unaire donnée, l'application $\mathfrak{D} \mapsto \llbracket \mathfrak{D}, \mathfrak{U}\mathfrak{D}_{\mathfrak{D}} \rrbracket$ est injective.

Démonstration: On prouve par récurrence que les n premiers niveaux du dessein sont calculables à partir de $\llbracket \mathfrak{D}, \mathfrak{U}\mathfrak{D}_{\mathfrak{D}} \rrbracket$. ■

Proposition 3.21 – Séparation

| La ludique probabiliste est séparée : pour tous desseins \mathfrak{D}_1 et \mathfrak{D}_2 sur une même base unaire, on a :

$$\mathfrak{D}_1 = \mathfrak{D}_2 \Leftrightarrow \forall \mathfrak{E}, \llbracket \mathfrak{D}_1, \mathfrak{E} \rrbracket = \llbracket \mathfrak{D}_2, \mathfrak{E} \rrbracket$$

Démonstration: Le point important, c'est que pour tout dessein \mathfrak{D} , l'opposant universel $\mathfrak{U}\mathfrak{D}_{\mathfrak{D}}$ dépend uniquement des interactions de \mathfrak{D} . Soit $\mathfrak{D}_1 \neq \mathfrak{D}_2$ deux desseins sur une même base unaire.

- Si $\text{Scal}^*(\mathfrak{D}_1) \neq \text{Scal}^*(\mathfrak{D}_2)$ alors $\llbracket \mathfrak{D}_1, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(1) \rrbracket \neq \llbracket \mathfrak{D}_2, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(1) \rrbracket$ d'après la proposition 3.18 page précédente.
- Si $\text{Scal}^*(\mathfrak{D}_1) = \text{Scal}^*(\mathfrak{D}_2)$ alors $\mathfrak{U}\mathfrak{D}_{\mathfrak{D}_1} = \mathfrak{U}\mathfrak{D}_{\mathfrak{D}_2}$. La proposition 3.20 permet de conclure. ■

Remarque 3.18 *Pour séparer deux desseins, il suffit donc de deux opposants : un premier pour calculer les scalaires et éventuellement un deuxième pour séparer les desseins si cela ne suffit pas.*

On peut aussi noter qu'on n'a pas la condition

$$\mathfrak{D}_1 = \mathfrak{D}_2 \Leftrightarrow (\forall \mathfrak{E}, \mathfrak{D}_1 \perp \mathfrak{E} \iff \mathfrak{D}_2 \perp \mathfrak{E})$$

Ainsi, par exemple, les desseins non immédiatement totaux (définition 3.17 page 81) n'ont pas d'orthogonaux.

Cette proposition peut être renforcée.

Proposition 3.22 – Séparation par des desseins universels fixés

| Soit α et α' deux scalaires distincts dans $]0; 1[$. Soit \mathfrak{D}_1 et \mathfrak{D}_2 deux desseins de base unaire β .

| Les desseins \mathfrak{D}_1 et \mathfrak{D}_2 sont égaux si et seulement si :

$$\llbracket \mathfrak{D}_1, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha) \rrbracket = \llbracket \mathfrak{D}_2, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha) \rrbracket \wedge \llbracket \mathfrak{D}_1, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha') \rrbracket = \llbracket \mathfrak{D}_2, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha') \rrbracket$$

Démonstration : Soit \mathfrak{D} un dessein de base β .

Les coefficients $c = \llbracket \mathfrak{D}, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha) \rrbracket$ et $c' = \llbracket \mathfrak{D}, \mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha') \rrbracket$ sont identiques à part les scalaires provenant des desseins universels qui sont multipliés par α'/α .

Or α'/α est différent de 1 donc dans c , il est possible de repérer les scalaires provenant de \mathfrak{D} (les scalaires identiques dans c') des scalaires provenant du dessein universel. C'est l'unique ingrédient qui suffit à calculer \mathfrak{D} (dans la proposition 3.20 page précédente) : on peut calculer par récurrence chaque niveau de \mathfrak{D} . ■

Remarque 3.19 *La proposition précédente montre que les desseins de base β sont séparés par toute paire de desseins universels distincts sur la base β^\perp . La proposition 3.21 page ci-contre montrait simplement qu'il existait une paire constructible en un certain sens ($\mathfrak{U}\mathfrak{D}_{\beta^\perp}(1)$ et $\mathfrak{U}\mathfrak{D}_{\mathfrak{D}}$) qui séparait.*

Intuitivement, on peut aussi rapprocher cette idée de paire de scalaires avec des coefficients complexes. Prendre deux scalaires différents α et α' et considérer les coefficients c et c' revient à plonger les scalaires s de \mathfrak{D} dans $s(1+i)$ et à considérer le coefficient résultant de la normalisation du réseau \mathfrak{D} , $\mathfrak{U}\mathfrak{D}_{\beta^\perp}(\alpha + \alpha' * i)$. Les scalaires du dessein universel (de la forme $2^{-n}(\alpha + \alpha' * i)$) sont différents des $s*(1+i)$ car α et α' sont différents.*

Elle prouve aussi au passage qu'il est possible de restreindre l'espace des scalaires à des ensembles beaucoup plus petits que l'intervalle $]0; 1[$ comme par exemple les rationnels dans $]0; 1[$ ou même une famille $(a_n)_{n \in \mathbb{N}}$ où les (a_n) sont différents et majorés par $\frac{1}{2}$ pour une infinité d'entre eux.

3.5.2 Stabilité

La stabilité stipule que l'intersection d'un ensemble de réseaux *compatibles* normalise en l'intersection des normalisations de ces réseaux. La bonne notion d'intersection est celle d'intersection pleine.

Définition 3.51 – Compatibilité pleine

Deux desseins sont **pleinement compatibles** lorsqu'ils sont pleinement inclus dans un même dessein.

Proposition 3.23 – Compatibilité et normalisation

Soit $\mathfrak{R} \subseteq \mathfrak{R}'$. On a

$$\llbracket \mathfrak{R} \rrbracket \subseteq \llbracket \mathfrak{R}' \rrbracket$$

Démonstration : On utilise le codage. Soient $\mathfrak{R}_1 \subseteq \mathfrak{R}'_1$ deux réseaux. Soient \mathfrak{R}_2 et \mathfrak{R}'_2 leurs traduits en ludique simple. Par monotonie de la traduction, on a $\mathfrak{R}_2 \subseteq \mathfrak{R}'_2$. Par le théorème de monotonie de la ludique simple (proposition 1.5 page 36) $\llbracket \mathfrak{R}_2 \rrbracket \subseteq \llbracket \mathfrak{R}'_2 \rrbracket$ et donc $\llbracket \mathfrak{R}_1 \rrbracket \subseteq \llbracket \mathfrak{R}'_1 \rrbracket$ par monotonie de la traduction inverse (proposition 3.8 page 95). ■

L'inclusion pleine permet de définir une notion d'intersection : l'intersection pleine.

Proposition 3.24 – Intersection pleine

Pour toute famille (\mathfrak{D}_i) de desseins compatibles, l'ensemble formé des c -chroniques pleinement incluses dans chaque \mathfrak{D}_i décrit un dessein.

Démonstration : Soit X l'ensemble formé des c -chroniques pleinement incluses dans chaque \mathfrak{D}_i . Il suffit de remarquer que deux c -chroniques dans X sont cohérentes. On conclut par la proposition 3.12 page 98. ■

Définition 3.52 – Intersection pleine

L'**intersection pleine** d'une famille (\mathfrak{D}_i) de desseins compatibles $\bigcap_i \mathfrak{D}_i$ est le dessein formé des c -chroniques pleinement incluses dans chaque \mathfrak{D}_i .

Théorème 3.25 – Stabilité

Soit (\mathfrak{R}_i) une famille de réseaux inclus dans un réseau \mathfrak{R} . On a

$$[[\bigcap \mathfrak{R}_i]] = \bigcap [[\mathfrak{R}_i]]$$

Démonstration : Il faut d'abord vérifier que cette définition a bien un sens c'est-à-dire que les desseins $[[\mathfrak{R}_i]]$ sont bien compatibles. C'est vrai par la Proposition 3.23 page précédente.

On utilise la normalisation par codage (définie section 3.3.1 page 82) ce qui permet de relever le théorème de stabilité de la ludique simple au cas probabiliste. On ne peut directement utiliser la proposition 3.10 page 96 car cette proposition ne prend pas en compte l'intersection. ■

Remarque 3.20 La définition d'inclusion pleine semble nécessaire à ce théorème car c'est celle qui permet de bien localiser un dessein inclus dans un autre. Une version plus relâchée (on autorise par exemple à rajouter des branches aux coefficients) ne semble pas marcher aussi naturellement.

3.5.3 Associativité

On réutilise la définition 1.35 page 36 des réseaux de réseaux dans le cadre probabiliste.

Définition 3.53 – Réseau de réseaux

Un **réseau de réseaux** $\{\mathfrak{R}_1, \dots, \mathfrak{R}_n\}$ est une partition d'un réseau en réseaux.

Proposition 3.26 – Associativité

Pour un réseau de réseaux $\{\mathfrak{R}_1, \dots, \mathfrak{R}_n\}$, on a

$$[[\mathfrak{R}_1, \dots, \mathfrak{R}_n]] = [[[\mathfrak{R}_1], \dots, [\mathfrak{R}_n]]]$$

Démonstration : On utilise le principe des équations génériques (proposition 3.10 page 96). ■

3.5.4 Monotonie

Proposition 3.27 – Monotonie

La normalisation est croissante vis-à-vis de l'ordre extensionnel \preceq .

$$\forall i, \mathcal{D}_i \preceq \mathcal{D}'_i \implies \llbracket \mathcal{D}_1, \dots, \mathcal{D}_n \rrbracket \preceq \llbracket \mathcal{D}'_1, \dots, \mathcal{D}'_n \rrbracket$$

Démonstration : On le vérifie par traduction en ludique simple : l'ordre extensionnel commute avec la traduction. Ce résultat peut être compris comme une sorte de *principe des inéquations génériques*. ■

Chapitre 4

Ludique probabiliste : à propos d'orthogonalité

Sommaire

4.1	Une caractérisation de l'orthogonalité	114
4.1.1	Desseins quasi-simples	115
4.1.2	Couvertures	115
4.1.3	Développement d'un dessein fini	118
4.1.4	Caractérisation	124
4.2	Plongements de la ludique simple	125

L'objectif de ce chapitre est d'étudier l'orthogonalité en ludique probabiliste de manière un peu plus précise. Des phénomènes discontinus apparaissent, typiques du passage à l'infini. La première section est consacrée à une caractérisation de l'orthogonalité par une technique d'approximation. La deuxième section décrit une famille de plongements de la ludique simple en ludique probabiliste.

4.1 Une caractérisation de l'orthogonalité

Cette section caractérise l'orthogonalité en ludique probabiliste. L'outil principal est la notion de *couverture* (définition 4.6 page 116) qui permet de "considérer" une partie finie d'un dessein probabiliste comme un ensemble de desseins simples muni d'une "probabilité".

On ne considère que des desseins complets car un branchement



n'est pas visité dans une interaction convergente (le poids de la normalisation est 1) si la somme des s_i est différente de 1 (proposition 5.4 page 132).

On commence par un petit lemme sur les suites réelles de somme proche de 1.

Proposition 4.1 – Suites réelles de somme proche de 1

Soit (a_i) et (b_i) deux suites finies de réels dans $[0; 1]$ de même longueur n .

Soit $\epsilon \in]0; 1]$. Si

$$- \sum a_i b_i \geq 1 - \epsilon^2$$

$$- \sum a_i \leq 1$$

alors il existe I tel que

$$\sum_{i \in I} a_i \geq 1 - \epsilon \text{ et } \forall i \in I, b_i \geq 1 - \epsilon$$

Démonstration : Soit $I = \{i \mid b_i \geq 1 - \epsilon\}$. Soit $A = \sum_{i \in I} a_i$. On a

$$1 - \epsilon^2 \leq \sum a_i b_i = \sum_{i \notin I} a_i b_i + \sum_{i \in I} a_i b_i \leq (1 - \epsilon)(1 - A) + A = 1 - \epsilon + \epsilon A$$

donc

$$-\epsilon^2 \leq -\epsilon + \epsilon A$$

donc

$$A \geq \frac{\epsilon - \epsilon^2}{\epsilon} = 1 - \epsilon$$

■

4.1.1 Desseins quasi-simples

Pour approximer un dessein probabiliste avec un ensemble de desseins simples, il est utile de savoir parler de dessein simple “modulo” scalaires. L'outil utile est la notion de dessein quasi-simple (définition 4.2).

Définition 4.1 – Coefficient quasi-simple

Les coefficients **quasi-simples** (finis ou infinis) sont engendrés par la grammaire suivante :

$$c := \smile \mid (s.\mathbf{0}, \dots, s.\mathbf{0}, s.c, s.\mathbf{0}, \dots, s.\mathbf{0})$$

L'idée sous-jacente à cette définition est qu'un coefficient quasi-simple inclus dans un coefficient c représente une branche de c .

Définition 4.2 – Dessein quasi-simple

Un **dessein quasi-simple** est un dessein dont chaque coefficient est quasi-simple.

Définition 4.3 – Oubli simple d'un dessein quasi-simple

L'**oubli simple** $\text{Os}(\mathfrak{D})$ d'un dessein quasi-simple \mathfrak{D} est le dessein obtenu en enlevant les coefficients et en gardant l'unique continuation de chaque coefficient.

Définition 4.4 – Poids d'un dessein quasi-simple

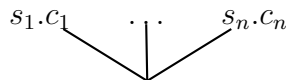
Le **poids** $\mathcal{W}(\mathfrak{D})$ d'un dessein quasi-simple \mathfrak{D} est le produit (éventuellement infini) des scalaires apparaissant dans ses coefficients.

Remarque 4.1 Le produit d'un ensemble infini X est la borne inférieure des produits des sous-familles finies de X . Naturellement, un produit infini est nul si la borne inférieure des produits des sous-familles finies est nulle.

4.1.2 Couvertures

Définition 4.5 – Choix dans un dessein

Un **choix** dans un dessein est la position d'un coefficient



Deux desseins quasi-simples \mathfrak{D}_1 et \mathfrak{D}_2 pleinement inclus dans un même dessein \mathfrak{D} font un **choix incompatible** s'il existe un choix de \mathfrak{D} tel que \mathfrak{D}_1 et \mathfrak{D}_2 contiennent des branches distinctes de ce choix (\mathfrak{D}_1 contient un certain s_{i_1} et \mathfrak{D}_2 un certain s_{i_2} avec $i_1 \neq i_2$).

La notion suivante de couverture est la notion fondamentale de ce chapitre. Elle permet de décrire une famille de desseins simples approximant un dessein probabiliste.

Définition 4.6 – Couverture

Soit β une base unaire. Soit \mathcal{D} un dessein probabiliste de base β .

Une **couverture** \mathcal{C} de \mathcal{D} est un ensemble fini de desseins quasi-simples finis inclus dans \mathcal{D} tels que si $X \in \mathcal{C}$ et $X' \in \mathcal{C}$ avec $X \neq X'$ alors ils font au moins un choix incompatible.

Soit \mathfrak{E} un dessein de base β^\perp . Une (α, β) -**couverture face à \mathfrak{E}** est une couverture \mathcal{C} de \mathcal{D} telle que

- $\forall X \in \mathcal{C}, \mathcal{W}(\llbracket \text{Os}(X), \mathfrak{E} \rrbracket) \geq \alpha$
- $\sum_{X \in \mathcal{C}} \mathcal{W}(X) \geq \beta$

Remarque 4.2 Étant donné un dessein \mathcal{D} , l'intuition d'une (α, β) -couverture de \mathcal{D} face à \mathfrak{E} est une approximation \mathcal{D} à $(1 - \beta)$ près orthogonale à \mathfrak{E} à $(1 - \alpha)$ près.

Définition 4.7 – Dessein décrit par une couverture

Soit \mathcal{D} un dessein et \mathcal{C} une couverture de \mathcal{D} . Le **dessein $\mathcal{D}_{\mathcal{C}}$ couvert par \mathcal{C}** est le plus petit dessein inclus dans \mathcal{D} contenant \mathcal{C} .

Définition 4.8 – Ordre sur les couvertures et couverture maximale

Les couvertures sur un dessein \mathcal{D} peuvent être munies de l'**ordre d'inclusion**.

Une couverture \mathcal{C} est **maximale** (pour $\mathcal{D}_{\mathcal{C}}$) si pour toute couverture \mathcal{C}' contenant strictement \mathcal{C} ,

$$\mathcal{D}_{\mathcal{C}} \subset \mathcal{D}_{\mathcal{C}'}$$

Proposition 4.2 – Existence de couverture maximale

Pour toute couverture \mathcal{C} , il existe une couverture maximale pour $\mathcal{D}_{\mathcal{C}}$ contenant \mathcal{C} .

Démonstration: Il n'y a qu'un ensemble fini de couvertures pour $\mathcal{D}_{\mathcal{C}}$ contenant \mathcal{C} . ■

Proposition 4.3 – Couverture

Soit \mathcal{C} une couverture pour un dessein complet \mathcal{D} (c'est-à-dire $\mathcal{D}_{\mathcal{C}} = \mathcal{D}$). On a :

- tout sous-ensemble de \mathcal{C} est une couverture ;
- $\sum_{X \in \mathcal{C}} \mathcal{W}(X) \leq 1$;
- si \mathcal{C} est maximale alors $\sum_{X \in \mathcal{C}} \mathcal{W}(X) = 1$.

Démonstration: Le premier point est clair.

Par croissance de

$$\mathcal{C} \mapsto \sum_{X \in \mathcal{C}} \mathcal{W}(X)$$

et existence d'une couverture maximale, il suffit de montrer le troisième point.

Soit \mathcal{C} une couverture maximale. Par simplicité, et sans perte de généralité (car \mathcal{D} est complet), on suppose les choix binaires $(p_i, 1 - p_i)_{i \in \{1, \dots, n\}}$.

On note $p_i^{\epsilon_i} = \begin{cases} p_i & \text{si } \epsilon = 1 \\ 1 & \text{si } \epsilon = 0 \\ 1 - p_i & \text{si } \epsilon = -1 \end{cases}$

On identifie les choix à des variables polynomiales. Soit E l'ensemble des polynômes à n variables (p_i) . Soit $X \in \mathcal{C}$. Le produit des choix de X est un polynôme en les (p_i) .

Soit F l'ensemble des polynômes ainsi associés aux éléments de la couverture \mathcal{C} .

Soit $P \in F$. Le polynôme P s'écrit

$$P = p_1^{\epsilon_1^P} * \dots * p_n^{\epsilon_n^P}$$

On réécrit F en plusieurs étapes à l'aide de la transformation suivante : si $\epsilon_i^P = 0$, on remplace P dans F par les deux polynômes $P * p_i$ et $P * (1 - p_i)$. Cette opération termine car elle augmente le degré (qui est borné). Soit F' l'ensemble de polynômes obtenus.

Par récurrence, la somme des polynômes P de F est égale à celle des polynômes P de F' .

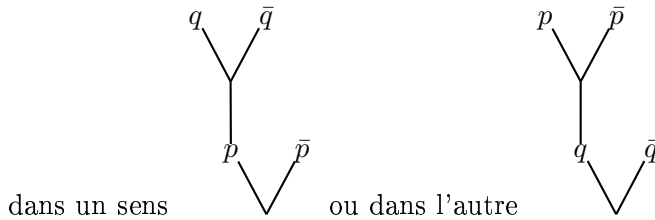
Si $P \in F'$ alors P s'écrit $P = p_1^{\epsilon_1^P} * \dots * p_n^{\epsilon_n^P}$ avec pour tout i , $\epsilon_i^P \neq 0$.

Par récurrence sur n , avec la maximalité de \mathcal{C} , on obtient que pour tout $i \in \{1, \dots, n\}$ si $P = p_1^{\epsilon_1^P} * \dots * p_n^{\epsilon_n^P} \in F'$ alors, pour tout i_0 , $p_1^{\epsilon_{i_0}^P} * \dots * p_n^{\epsilon_n^P} \in F'$

où $\epsilon_i^P = \begin{cases} -\epsilon_i^P & \text{si } i = i_0 \\ \epsilon_i^P & \text{sinon} \end{cases}$

Cette récurrence repose sur la remarque suivante. Deux choix (p, \bar{p}) et (q, \bar{q}) peuvent être positionnés :

- soit séquentiellement (un choix dépend d'un autre choix en ce sens qu'il n'est possiblement présent dans un $X \in \mathcal{C}$ que si un autre choix est dans X)



- soit parallèlement

Cette notion s'étend aux ensembles de polynômes. Dans ce cadre, le passage de F à F' revient à paralléliser tous les choix entre eux.

On en conclut immédiatement que

$$\sum_{P \in F'} P = 1$$

donc

$$\sum_{P \in F} P = 1$$

et enfin

$$\sum_{X \in \mathcal{C}} \mathcal{W}(X) = 1 \quad \blacksquare$$

Remarque 4.3 Dans la preuve de la proposition précédente, l'ensemble de polynômes F' ne correspond pas à une couverture : on utilise un oubli de la structure d'arbre.

Proposition 4.4 – Existence de couverture maximale

Soit \mathcal{D}_0 un dessein fini inclus dans un dessein \mathcal{D} . Il existe une couverture maximale \mathcal{C} de \mathcal{D} telle que $\mathcal{D}_{\mathcal{C}} = \mathcal{D}_0$.

Démonstration : Soit \mathcal{C} l'ensemble des desseins quasi-simples maximaux inclus dans \mathcal{D}_0 .

Par maximalité de chaque dessein quasi-simple, l'ensemble \mathcal{C} est bien une couverture (les desseins quasi-simples sont incompatibles donc maximaux).

La couverture \mathcal{C} est bien maximale car tout dessein quasi-simple inclus dans \mathcal{D}_0 est inclus dans un dessein quasi-simple de \mathcal{C} . ■

4.1.3 Développement d'un dessein fini

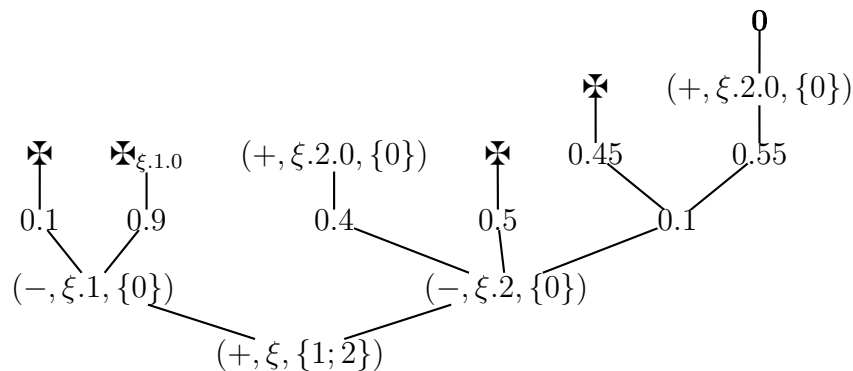
Les desseins développés sont les combinaisons linéaires des desseins simples. Ils jouent un rôle important pour les couvertures (proposition 4.7 page 123).

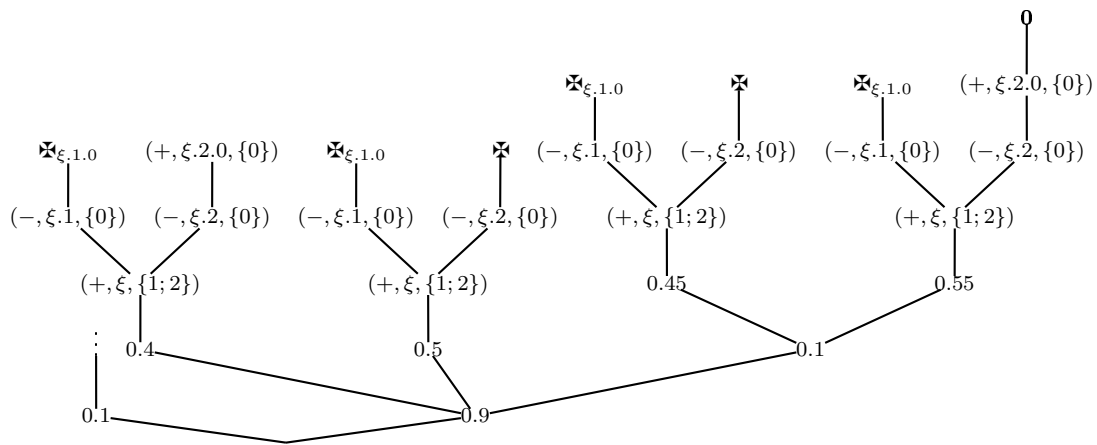
Définition 4.9 – Dessein développé

Un dessein **développé** est un dessein de la forme $c.(\mathcal{D}_1, \dots, \mathcal{D}_n)$ où les (\mathcal{D}_i) sont simples.

Remarque 4.4 Dans la définition précédente, la convention utilisée est encore d'accepter les combinaisons linéaires d'une infinité de desseins notées $c.(\mathcal{D}_1, \dots, \mathcal{D}_n)$.

La définition suivante donne un moyen de passer d'un dessein fini à un dessein fini développé. Pour la motiver, un exemple de dessein et de dessein développé est d'abord donné :

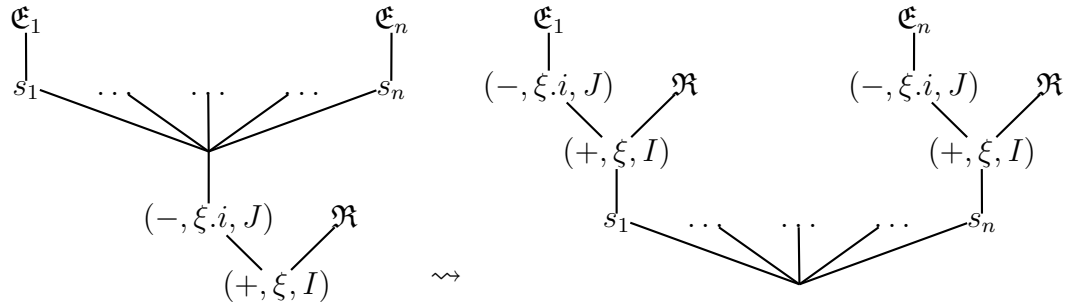




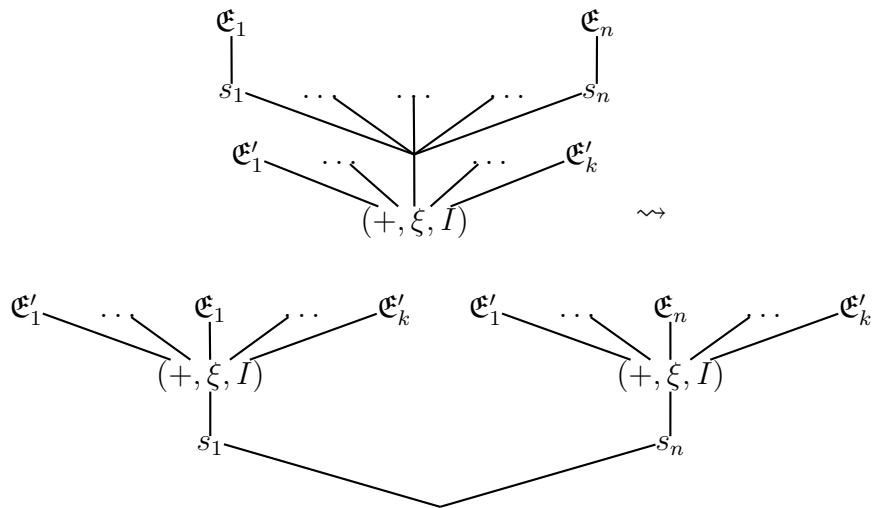
Définition 4.10 – Développement d'un dessin fini

Un **développement** d'un dessin fini complet \mathcal{D} est une forme normale à partir de \mathcal{D} pour le système de réécriture formé

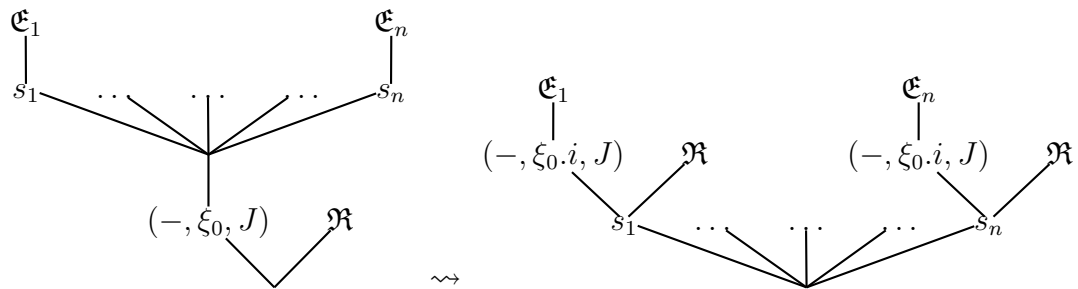
– de la règle



– de la règle



– ainsi que, si \mathcal{D} est négatif sur la base $\xi_0 \vdash \Lambda$, de la règle



Remarque 4.5 Soit h la hauteur du dessin \mathcal{D} et n le nombre de choix de \mathcal{D} . Le système de réécriture utilisé dans la définition précédente termine car

- la hauteur de chaque réduit est bornée par $h + n$ et
- à chaque étape, la suite du nombre d'actions aux hauteurs $1, 2, \dots, h, \dots, h + n$ décroît strictement suivant l'ordre lexicographique.

De plus, un développement d'un dessin fini est bien un dessin développé.

Proposition 4.5 – Conservation du poids par développement

Pour tout développement \mathcal{D}' d'un dessin fini complet \mathcal{D} , on a pour tout \mathcal{E} de base orthogonale :

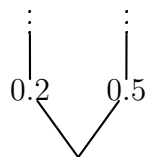
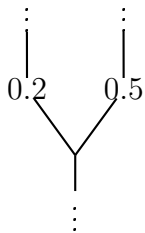
$$\mathcal{W}([\mathcal{D}, \mathcal{E}]) = \mathcal{W}([\mathcal{D}', \mathcal{E}])$$

Démonstration : On le vérifie à chaque étape du développement. Le point crucial est que pour tout branchement



la somme des s_i est égale à 1. ■

Remarque 4.6 L'hypothèse de complétude du dessin est cruciale. En effet, considérons un dessin positif \mathcal{D} orthogonal à un dessin \mathcal{E} . Si \mathcal{D} est de la forme



un développement de \mathcal{D} est et n'est pas orthogonal à \mathcal{E} (le poids de toute interaction de \mathcal{D} avec un autre dessin est borné par $0.2 + 0.5 = 0.7$).

À une projection simple dans \mathcal{D} correspond plusieurs projections simples, les *résidus*.

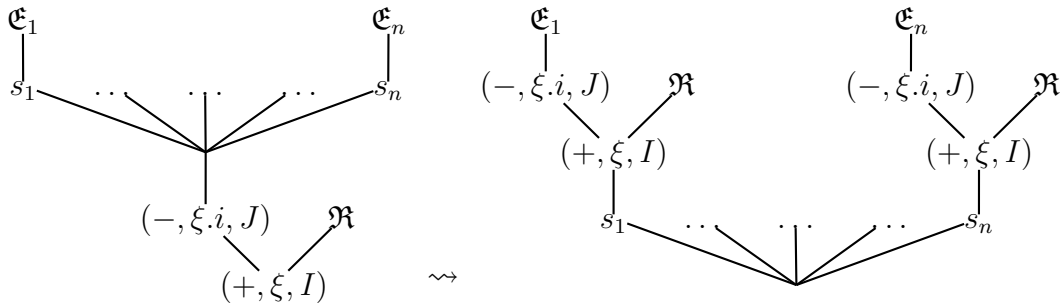
Définition 4.11 – Résidus par développement

Soit \mathfrak{D} un dessin fini complet. Un **développement en séquence** de \mathfrak{D} est une suite

$$\mathfrak{D} = \mathfrak{D}_1 \rightsquigarrow \dots \rightsquigarrow \mathfrak{D}_n = \mathfrak{D}' \not\rightsquigarrow$$

Les résidus $r_i(X)$ d'un dessin quasi-simple X de \mathfrak{D}_i à travers une étape $\mathfrak{D}_i \rightsquigarrow \mathfrak{D}_{i+1}$ sont définis ainsi : (on ne considère que la première règle, les autres règles sont similaires).

Soit X un dessin quasi-simple de \mathfrak{D}_i .



Les résidus de X sont les dessins quasi-simples X' de \mathfrak{D}_{i+1} tels que

- X et X' sont égaux sur la partie en dehors du motif.
- si X n'intersecte pas le choix



mais intersecte $\begin{matrix} \mathfrak{R} \\ | \\ (+, \xi, I) \end{matrix}$ alors X' intersecte de la même manière un des $\begin{matrix} \mathfrak{R} \\ | \\ (+, \xi, I) \end{matrix}$ sans intersecter les actions $(-, \xi, i, J)$.

- si X intersecte \mathfrak{E}_{j_0} , alors X' intersecte de la même manière \mathfrak{E}_{j_0} et de même pour

la partie $\begin{matrix} \mathfrak{R} \\ | \\ (+, \xi, I) \end{matrix}$.

Les résidus $R_i(X)$ d'un ensemble de dessins quasi-simples $\{X_1, \dots, X_m\}$ sont définis par union : $\cup_j r_i(X_j)$.

Les résidus $R(X)$ d'un dessin quasi-simple X de \mathfrak{D} à travers toutes les étapes de développement sont

$$R_n(R_{n-1}(\dots(R_1(\{X\})))\dots))$$

Proposition 4.6 – Résidus

- Les résidus d'un dessein quasi-simple ont même oubli simple.
- De plus, la somme des poids des résidus d'un dessein quasi-simple est égale au poids du dessein.
- Pour tout \mathfrak{E} et tout \mathfrak{D} , $\mathcal{W}(\llbracket \mathfrak{D}, \mathfrak{E} \rrbracket) = \sum_{\mathfrak{D}' \in R(\mathfrak{D})} \mathcal{W}(\llbracket \mathfrak{D}', \mathfrak{E} \rrbracket)$.

Démonstration : On le vérifie à chaque étape de développement. ■

Définition 4.12 – Couverture développée

Une couverture \mathcal{C} d'un dessein \mathfrak{D} est **développée** si le dessein $\mathfrak{D}_{\mathcal{C}}$ est développé.

Proposition 4.7 – Couverture d'un dessein développé

Soient \mathfrak{D} un dessein fini complet développé et \mathfrak{E} un dessein sur une base orthogonale. Soit \mathcal{C} une couverture développée pour \mathfrak{D} . On a :

- $\mathcal{W}(\llbracket \mathfrak{D}_{\mathcal{C}}, \mathfrak{E} \rrbracket) = \sum_{X \in \mathcal{C}} \mathcal{W}(\llbracket X, \mathfrak{E} \rrbracket)$
- $\forall X \in \mathcal{C}$, $\mathcal{W}(\llbracket X, \mathfrak{E} \rrbracket) = \mathcal{W}(\llbracket \text{Os}(X), \mathfrak{E} \rrbracket) \mathcal{W}(X)$

Démonstration :

On montre le cas où \mathfrak{D} est positif.

- Le dessein $\mathfrak{D}_{\mathcal{C}}$ est aussi positif et s'écrit $\mathfrak{D}_{\mathcal{C}} = c.(\text{Os}(X_1), \dots, \text{Os}(X_n))$.
- On a

$$\llbracket \mathfrak{D}_{\mathcal{C}}, \mathfrak{E} \rrbracket = c.(\llbracket \text{Os}(X_1), \mathfrak{E} \rrbracket, \dots, \llbracket \text{Os}(X_n), \mathfrak{E} \rrbracket)$$

- C'est clair ($X \in \mathcal{C}$ est essentiellement une branche de scalaires suivie d'un dessein simple).

Le cas où \mathfrak{D} est négatif est un petit peu plus compliqué : on procède de la même manière mais à chaque feuille vide du premier coefficient de \mathfrak{E} . ■

Proposition 4.8 – Couverture développée

Soit \mathfrak{D} un dessein fini complet. Soit \mathcal{C} une (α, β) -couverture sur \mathfrak{D} . Il existe \mathcal{C}^{dev} et $\mathfrak{D}_{\mathcal{C}}^{dev}$ tels que :

- Le dessein $\mathfrak{D}_{\mathcal{C}}^{dev}$ est la cible d'un développement en séquence de $\mathfrak{D}_{\mathcal{C}}$. Soit R la fonction de résidus associée à ce développement en séquence.
- \mathcal{C}^{dev} est une (α, β) -couverture sur $\mathfrak{D}_{\mathcal{C}}^{dev}$.
- Si $X \in \mathcal{C}$, alors les résidus de X sont dans \mathcal{C}^{dev} .
- Pour tout \mathfrak{E} , $\mathcal{W}(\llbracket \mathfrak{D}_{\mathcal{C}}, \mathfrak{E} \rrbracket) = \mathcal{W}(\llbracket \mathfrak{D}_{\mathcal{C}}^{dev}, \mathfrak{E} \rrbracket)$.
- $\sum_{X \in \mathcal{C}} \mathcal{W}(X) = \sum_{X \in \mathcal{C}^{dev}} \mathcal{W}(X)$.
- Pour tout \mathfrak{E} et tout $X \in \mathcal{C}$, $\mathcal{W}(\llbracket X, \mathfrak{E} \rrbracket) = \sum_{X' \in R(X)} \mathcal{W}(\llbracket X', \mathfrak{E} \rrbracket)$.
- Si \mathcal{C} est maximale, alors \mathcal{C}^{dev} est aussi maximale.

On note $\text{Dev}(\mathcal{C}) = \mathcal{C}^{dev}$.

Démonstration : On construit la couverture \mathcal{C}^{dev} par étapes durant le développement de \mathfrak{D} en prenant les résidus de la couverture précédente. C'est encore une couverture (la condition d'incompatibilité est bien préservée). De plus, c'est une couverture sur le dessein réduit.

Il reste à prouver

- \mathcal{C}^{dev} est une (α, β) -couverture.
C'est vrai par la proposition 4.6 page précédente.
- Pour tout \mathfrak{E} , $\mathcal{W}(\llbracket \mathfrak{D}_{\mathcal{C}}, \mathfrak{E} \rrbracket) = \mathcal{W}(\llbracket \mathfrak{D}_{\mathcal{C}^{dev}}, \mathfrak{E} \rrbracket)$.
C'est vrai par la proposition 4.5 page 121.
- $\sum_{X \in \mathcal{C}} \mathcal{W}(X) = \sum_{X \in \mathcal{C}^{dev}} \mathcal{W}(X)$.
C'est vrai par la proposition 4.6 page précédente.
- Pour tout \mathfrak{E} et tout $X \in \mathcal{C}$, $\mathcal{W}(\llbracket X, \mathfrak{E} \rrbracket) = \sum_{X' \in R(X)} \mathcal{W}(\llbracket X', \mathfrak{E} \rrbracket)$.
C'est vrai par la proposition 4.6 page précédente.
- Si \mathcal{C} est maximale, alors \mathcal{C}^{dev} est aussi maximale.

Il suffit donc de montrer que si \mathcal{C} est maximale, alors \mathcal{C}^{dev} est aussi maximale. Pour cela, il suffit de montrer qu'une étape de développement $\mathfrak{D} \rightsquigarrow \mathfrak{D}'$ transforme une couverture maximale \mathcal{C} de \mathfrak{D} en une couverture maximale \mathcal{C}' de \mathfrak{D}' .

Supposons que \mathcal{C}' ne soit pas maximale. Soit $X'_\nu \notin \mathcal{C}'$ tel que $\mathcal{C}' \cup X'_\nu$ est une couverture de \mathfrak{D}' . On peut remonter X'_ν en arrière par le développement. $X'_\nu \notin \mathcal{C}'$ donc $X'_\nu \notin \mathcal{C}$. Par maximalité de \mathcal{C} , X'_ν est compatible avec un certain $X_0 \in \mathcal{C}$. Mais, alors, X'_ν est compatible avec les résidus de X_0 donc $\mathcal{C}' \cup X'_\nu$ n'est pas une couverture. Il y a contradiction donc \mathcal{C}' est maximale. ■

4.1.4 Caractérisation

Il est maintenant possible d'énoncer et de démontrer la proposition suivante qui stipule que deux desseins \mathfrak{D} et \mathfrak{E} sont orthogonaux si et seulement si on peut couvrir \mathfrak{D} face à \mathfrak{E} de manière aussi proche de 1 que désirée.

Proposition 4.9 – Caractérisation de l'orthogonalité

Soient \mathfrak{D} et \mathfrak{E} deux desseins complets. Les desseins \mathfrak{D} et \mathfrak{E} sont orthogonaux si et seulement si

$$\forall \epsilon \in]0; 1[, \exists \mathcal{C}, \text{ une } ((1 - \epsilon), (1 - \epsilon)) \text{ - couverture de } \mathfrak{D} \text{ face à } \mathfrak{E}$$

Démonstration : On montre chaque implication.

⇒ Supposons $\mathfrak{D} \perp \mathfrak{E}$.

Soit $\epsilon \in]0; 1[$.

Soit \mathfrak{D}_0 un dessein fini pleinement inclus dans \mathfrak{D} tel que $\mathcal{W}(\llbracket \mathfrak{D}_0, \mathfrak{E} \rrbracket) \geq 1 - \epsilon^2$.

Soit \mathcal{C} une couverture de \mathfrak{D} maximale telle que $\mathfrak{D}_{\mathcal{C}} = \mathfrak{D}_0$ (l'existence est assurée par la proposition 4.4 page 118).

Soit $\mathcal{C}^{dev} = \text{Dev}(\mathcal{C})$. On note $\mathfrak{D}_{\mathcal{C}^{dev}}^{dev} = \mathfrak{D}_0^{dev}{}_{\mathcal{C}^{dev}}$.

La proposition 4.8 page précédente implique que \mathcal{C}^{dev} est maximale.

On a, par maximalité de \mathcal{C}^{dev} ,

$$1 - \epsilon^2 \leq \mathcal{W}(\llbracket \mathfrak{D}_0, \mathfrak{E} \rrbracket) = \mathcal{W}(\llbracket \mathfrak{D}_{\mathcal{C}^{dev}}^{dev}, \mathfrak{E} \rrbracket) = \sum_{X \in \mathcal{C}^{dev}} \mathcal{W}(\llbracket \text{Os}(X), \mathfrak{E} \rrbracket) \mathcal{W}(X)$$

On a $\sum_{X \in \mathcal{C}^{dev}} \mathcal{W}(X) \leq 1$ donc, d'après la proposition 4.1 page 114, il existe une

couverture $\mathcal{C}_0 \subseteq \mathcal{C}^{dev}$ tel que

$$\sum_{X \in \mathcal{C}_0} \mathcal{W}(X) \geq 1 - \epsilon$$

et

$$\forall X \in \mathcal{C}_0, \mathcal{W}(\llbracket \text{Os}(X), \mathfrak{E} \rrbracket) \geq 1 - \epsilon$$

On en conclut que \mathcal{C}_0 est une $(1 - \epsilon, 1 - \epsilon)$ -couverture de $\mathfrak{D}_{\mathcal{C}}^{dev}$.

Soit $\mathcal{C}' = \{X \mid R(X) \cap \mathcal{C}_0 \neq \emptyset\}$ où R est la fonction de résidus du développement de \mathcal{C} et \mathcal{C}^{dev} .

On conclut en remarquant que \mathcal{C}' est une $(1 - \epsilon, 1 - \epsilon)$ -couverture de \mathfrak{D} car :

- $\forall X \in \mathcal{C}', \forall X' \in R(X), \mathcal{W}(\llbracket \text{Os}(X), E \rrbracket) = \mathcal{W}(\llbracket \text{Os}(X'), E \rrbracket) \geq 1 - \epsilon$
- $\sum_{X \in \mathcal{C}'} \mathcal{W}(X) \geq \sum_{X' \in \mathcal{C}_0} \mathcal{W}(X') \geq 1 - \epsilon$

⇐ Supposons

$$\forall \epsilon \in]0; 1[, \exists \mathcal{C}, \text{ une } (1 - \epsilon, 1 - \epsilon) \text{ - couverture de } \mathfrak{D} \text{ face à } \mathfrak{E}$$

Soit $\epsilon \in]0; 1[$. Soit \mathcal{C} une $(1 - \epsilon, 1 - \epsilon)$ -couverture de \mathfrak{D} face à \mathfrak{E} .

On considère un développement \mathcal{C}^{dev} . On note $\mathfrak{D}_{\mathcal{C}}^{dev} = \mathfrak{D}_{\mathcal{C}^{dev}}^{dev}$.

On a

$$\begin{aligned} \mathcal{W}(\llbracket \mathfrak{D}, \mathfrak{E} \rrbracket) &\geq \mathcal{W}(\llbracket \mathfrak{D}_{\mathcal{C}}, \mathfrak{E} \rrbracket) = \mathcal{W}(\llbracket \mathfrak{D}_{\mathcal{C}}^{dev}, \mathfrak{E} \rrbracket) \geq \sum_{X' \in \mathcal{C}^{dev}} \mathcal{W}(\llbracket X', \mathfrak{E} \rrbracket) = \\ &\sum_{X' \in \mathcal{C}^{dev}} \mathcal{W}(\llbracket \text{Os}(X'), \mathfrak{E} \rrbracket) \mathcal{W}(X') = \sum_{X \in \mathcal{C}} \mathcal{W}(\llbracket \text{Os}(X), \mathfrak{E} \rrbracket) \mathcal{W}(X) \geq (1 - \epsilon)^2 \end{aligned}$$

Ceci est vrai pour tout $\epsilon \in]0; 1[$ donc $\mathcal{W}(\llbracket \mathfrak{D}, \mathfrak{E} \rrbracket) = 1$ et $\mathfrak{D} \perp \mathfrak{E}$. ■

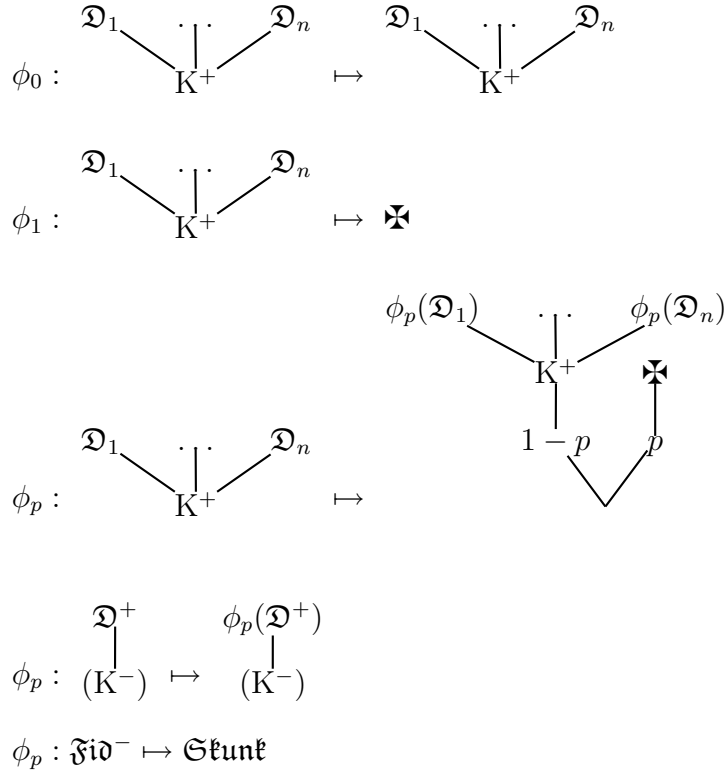
4.2 Plongements de la ludique simple

Cette section généralise le plongement canonique défini à la section 3.3.2 page 95. L'objectif est d'expliquer la structure et non de définir des choses techniquement utiles par la suite.

Le *plongement plus normalisant* défini dans cette section est une fonction des desseins simples dans les desseins probabilistes de même base qui retourne un dessin qui normalise avec plus de desseins que le dessin de départ.

Définition 4.13 – Plongement plus normalisant

La **fonction** ϕ_p des desseins simples d'une base donnée dans les desseins probabilistes de même base est définie par :



Proposition 4.10 – Orthogonalité et plongement plus normalisant

Soient \mathfrak{D} et \mathfrak{E} deux desseins sur des bases duales. On a :

$$\mathfrak{D} \perp \mathfrak{E} \Leftrightarrow \forall p, \mathcal{W}(\llbracket \phi_p(\mathfrak{D}), \mathfrak{E} \rrbracket) = 1$$

$$\mathfrak{D} \not\perp \mathfrak{E} \Leftrightarrow \forall p < 1, \mathcal{W}(\llbracket \phi_p(\mathfrak{D}), \mathfrak{E} \rrbracket) < 1$$

Le plongement plus normalisant ϕ_p est une notion raisonnable du point de vue de la ludique probabiliste car elle peut être internalisée comme le montrent la définition et la proposition suivante.

Définition 4.14 – Le fax plus normalisant

Le **fax plus normalisant** $\mathfrak{Fax}_{\xi+\xi'}^p$ est $\phi_p(\mathfrak{Fax}_{\xi+\xi'})$.

La proposition suivante remarque que la fonction ϕ_p est **définissable** en ludique probabiliste au sens que, modulo une normalisation avec $\mathfrak{Fax}_{\xi+\xi'}$ l'image par ϕ_p d'un dessin est son image par la normalisation avec le fax plus normalisant $\mathfrak{Fax}_{\xi+\xi'}^p$.

Proposition 4.11 – Définition interne de ϕ_p : le fax plus normalisant

Pour tout dessein \mathcal{D} sur une base $\vdash \xi$, on a

$$\llbracket \mathcal{D}, \mathfrak{Fax}_{\xi \vdash \xi'}^p \rrbracket = \phi_p(\llbracket \mathcal{D}, \mathfrak{Fax}_{\xi \vdash \xi'} \rrbracket)$$

Démonstration : Par récurrence sur n , on prouve par cas sur \mathcal{D} que l'égalité

$$\llbracket \mathcal{D}, \mathfrak{Fax}_{\xi \vdash \xi'}^p \rrbracket = \phi_p(\llbracket \mathcal{D}, \mathfrak{Fax}_{\xi \vdash \xi'} \rrbracket)$$

est vraie jusqu'à la hauteur n . ■

La proposition suivante montre que, via la fonction ϕ_p , la ludique probabiliste sait différencier la divergence en temps fini de la divergence en temps infini.

Proposition 4.12 – Disputes infinies

Soient \mathcal{D} et \mathcal{E} deux desseins simples sur des bases duales tels que la dispute entre \mathcal{D} et \mathcal{E} soit de longueur $k \in \mathbb{N} \cup \{+\infty\}$ et $\llbracket \mathcal{D}, \mathcal{E} \rrbracket = \mathfrak{FiD}$. On a alors :

$$\mathcal{W}(\llbracket \phi_p(\mathcal{D}), \mathcal{E} \rrbracket) = (1 - (1 - p)^k)$$

Définition 4.15 – Dessein simple continu

Un dessein simple \mathcal{D} est **continu** si pour tout dessein simple \mathcal{E} , la fonction

$$p \mapsto \mathcal{W}(\llbracket \phi_p(\mathcal{D}), \mathcal{E} \rrbracket)$$

est continue en 0. Le dessein \mathcal{D} est **discontinu** dans le cas contraire.

Définition 4.16 – Chemin

Un **chemin** d'un dessein \mathcal{D} est la partie de \mathcal{D} visitée lors d'une interaction donnée.

La proposition suivante caractérise les desseins continus.

Proposition 4.13 – Dessesins simples ayant une interaction infinie

Soit \mathfrak{D} un dessein simple. Les propriétés suivantes sont équivalentes :

1. Le dessein \mathfrak{D} a une chronique infinie.
2. Le dessein \mathfrak{D} a un chemin infini.
3. Le dessein \mathfrak{D} a une tranche infinie.
4. Le dessein \mathfrak{D} est discontinu : il existe un dessein simple \mathfrak{E} tel que la fonction $p \mapsto \mathcal{W}(\llbracket \phi_p(\mathfrak{D}), \mathfrak{E} \rrbracket)$ est discontinue en 0.
5. Il existe un dessein simple \mathfrak{E} tel que $\mathcal{W}(\llbracket \phi_p(\mathfrak{D}), \mathfrak{E} \rrbracket)$ est égale à 0 si p est nul et 1 sinon.
6. Il existe un dessein simple \mathfrak{E} tel que $\mathcal{W}(\llbracket \mathfrak{D}, \phi_p(\mathfrak{E}) \rrbracket)$ est égale à 0 si p est nul et 1 sinon.
7. Il existe un dessein simple \mathfrak{E} tel que la fonction $p \mapsto \mathcal{W}(\llbracket \mathfrak{D}, \phi_p(\mathfrak{E}) \rrbracket)$ est discontinue en 0.

Démonstration :

- 1 \Rightarrow 2. C'est vrai car toute chronique est un chemin.
 2 \Rightarrow 3. C'est vrai car tout chemin est une tranche.
 3 \Rightarrow 1. Chaque tranche est à branchement fini, donc, d'après le lemme de Koenig, toute tranche infinie contient une chronique infinie.
 2 \Rightarrow 4 et 5. On suppose que \mathfrak{D} a un chemin infini. Soit \mathfrak{E} parcourant ce chemin infini. D'après la proposition 4.12 page précédente, pour tout p différent de 0, $\mathcal{W}(\llbracket \phi_p(\mathfrak{D}), \mathfrak{E} \rrbracket) = 1$ mais $\mathcal{W}(\llbracket \phi_0(\mathfrak{D}), \mathfrak{E} \rrbracket) = \mathcal{W}(\llbracket \mathfrak{D}, \mathfrak{E} \rrbracket) = 0$.
 4 \Rightarrow 2. Soit \mathfrak{E} tel que $p \mapsto \llbracket \phi_p(\mathfrak{D}), \mathfrak{E} \rrbracket$ soit discontinu en 0. Puisque tout polynôme est continu, d'après la proposition 4.12 page précédente et la proposition 4.10 page 126, alors la dispute entre \mathfrak{D} et \mathfrak{E} est infinie et \mathfrak{D} a un chemin infini.
 5 \Rightarrow 4. La fonction caractéristique de $]0, 1]$ est bien discontinue.
 5 \Leftrightarrow 6. Dans la dispute entre \mathfrak{D} et \mathfrak{E} , aucun démon n'est atteint car $\mathcal{W}(\llbracket \mathfrak{D}, \mathfrak{E} \rrbracket) = 0$ donc les disputes probabilistes entre $\phi_p(\mathfrak{D})$ et \mathfrak{E} d'une part et \mathfrak{D} et $\phi_p(\mathfrak{E})$ d'autre part sont égales au décalage près d'un coup pour les coefficients p .
 6 \Rightarrow 7. La fonction caractéristique de $]0, 1]$ est bien discontinue.
 7 \Rightarrow 6. On procède comme précédemment. ■

La proposition 4.12 montre un apport de la ludique probabiliste à la ludique simple. Interactivement, la ludique probabiliste est capable de calculer le nombre d'étapes (dans $\mathbb{N} \cup \{+\infty\}$) avant de diverger lorsqu'une action K^+ ne trouve pas d'action duale K^- . Ainsi, la ludique probabiliste distingue la divergence en temps fini de la divergence causée par un calcul infini sans forme normale de tête.

La proposition 4.13 montre, de plus, que la ludique probabiliste sait caractériser interactivement les desseins avec des chroniques infinies : ce sont les desseins discontinus.

Chapitre 5

Ludique probabiliste : comportements et connecteurs

Sommaire

5.1	Comportements	130
5.1.1	Définitions	130
5.1.2	Combinaisons linéaires	130
5.1.3	Incarnation	132
5.1.4	Superposition	134
5.1.5	Notions pour la complétude interne des comportements	142
5.2	Connecteurs	145
5.2.1	Décalages	145
5.2.2	Additifs	146
5.2.3	Multiplicatifs	146
5.2.4	Quantificateurs	148

Les comportements et connecteurs probabilistes généralisent les comportements et connecteurs simples. Les notions de la ludique simple comme celles d'incarnation ou d'éthique complète sont adaptées et de nouvelles notions comme celle de combinaison linéaire apparaissent.

5.1 Comportements

Par simplicité, on se limite au cas de la base unaire. Les comportements probabilistes sont définis de manière similaire à la ludique simple, mais leur géométrie est plus importante. Ils possèdent plusieurs générateurs pour différentes opérations : les desseins semi-simples pour les combinaisons linéaires (section 5.1.2) et l'incarnation comme en ludique simple pour l'ordre stable (section 5.1.3 page 132).

5.1.1 Définitions

Un dessin a un orthogonal non vide si et seulement si il est immédiatement total (définition 3.17 page 81). La notion d'éthique probabiliste est donc restreinte aux desseins immédiatement totaux.

Définition 5.1 – Éthique

Une **éthique** est un ensemble non vide de desseins immédiatement totaux de même base.

Définition 5.2 – Orthogonal d'une éthique

L'**orthogonal** d'une éthique est défini par

$$E^\perp = \{\mathfrak{F} \mid \forall \mathfrak{D} \in E, \mathcal{W}(\llbracket \mathfrak{F}, \mathfrak{D} \rrbracket) = 1\}$$

Définition 5.3 – Comportement

Un **comportement** est une éthique égale à son bi-orthogonal.

La clôture par bi-orthogonal d'un comportement implique d'autres clôtures comme la clôture par combinaison linéaire et par superposition (définition 5.10 page 135). Il est alors naturel de considérer les ensembles générateurs par combinaison linéaire ou superposition d'un comportement.

5.1.2 Combinaisons linéaires

L'utilisation de coefficients amène à considérer la notion de combinaison linéaire. Le résultat principal (proposition 5.3 page 132) est que les comportements sont clos par combinaison linéaire et que la partie semi-simple d'un comportement engendre le comportement.

Définition 5.4 – Partie semi-simple

La **partie semi-simple** d'un comportement G est l'ensemble des desseins semi-simples de G . On la note $\text{Pss}(G)$.

Définition 5.5 – Combinaison linéaire

L'ensemble $\text{Cl}(E)$ est l'ensemble des combinaisons linéaires de coefficient de poids 1 d'un ensemble de desseins E sur une base donnée.

Définition 5.6 – Projection semi-simple

Un dessein semi-simple \mathfrak{D}_0 est une **projection semi-simple** d'un dessein \mathfrak{D} si \mathfrak{D}_0 appartient à la famille (\mathfrak{D}_α) de desseins semi-simples dont \mathfrak{D} est la combinaison linéaire.

Dans la proposition suivante, on utilise la notation $\mathfrak{D} = c.(\mathfrak{D}_1, \dots, \mathfrak{D}_n)$ pour représenter une combinaison linéaire quelconque (la famille $(\mathfrak{D}_i)_i$ n'est pas forcément finie).

Proposition 5.1 – Projection semi-simple et orthogonalité

Soit $\mathfrak{D} = c.(\mathfrak{D}_1, \dots, \mathfrak{D}_n)$ un dessein de base unaire β et \mathfrak{E} un dessein de base orthogonale. Alors,

$$\mathfrak{D} \perp \mathfrak{E} \Leftrightarrow \mathcal{W}(c) = 1 \text{ et } \forall i, \mathfrak{D}_i \perp \mathfrak{E}$$

Démonstration : Il faut faire deux cas suivant la polarité de \mathfrak{D} .

– Cas 1. \mathfrak{D} est positif.

Alors

$$[\mathfrak{D}, \mathfrak{E}] = c.([\mathfrak{D}_1, \mathfrak{E}], \dots, [\mathfrak{D}_n, \mathfrak{E}])$$

Par la proposition 3.4 page 74, on conclut que $\mathfrak{D} \perp \mathfrak{E}$ si et seulement si c est de poids 1 et que les $[\mathfrak{D}_i, \mathfrak{E}]$ sont aussi de poids 1.

– Cas 2. \mathfrak{D} est négatif.

On écrit $\mathfrak{E} = c'.(\mathfrak{E}_1, \dots, \mathfrak{E}_k)$ où les \mathfrak{E}_i sont semi-simples (on utilise toujours la convention que la famille des (\mathfrak{E}_i) peut être infinie).

On a

$$[\mathfrak{D}, \mathfrak{E}] = c'.([\mathfrak{D}, \mathfrak{E}_1], \dots, [\mathfrak{D}, \mathfrak{E}_k])$$

Donc

$$[\mathfrak{D}, \mathfrak{E}] = c'.(c.([\mathfrak{D}_1, \mathfrak{E}_1], \dots, [\mathfrak{D}_n, \mathfrak{E}_1]), \dots, c.([\mathfrak{D}_1, \mathfrak{E}_k], \dots, [\mathfrak{D}_n, \mathfrak{E}_k]))$$

Par la proposition 3.4 page 74, on déduit que $\mathfrak{D} \perp \mathfrak{E}$ si et seulement si c' est de poids 1, c aussi et que pour tout i et j , $\mathfrak{D}_i \perp \mathfrak{E}_j$.

Il reste à prouver pour tout i , que $\mathfrak{D}_i \perp \mathfrak{E}$ si et seulement si c' est de poids 1 et pour tout j , $\mathfrak{D}_i \perp \mathfrak{E}_j$.

Fixons i . On a

$$[\mathfrak{D}_i, \mathfrak{E}] = c'.([\mathfrak{D}_i, \mathfrak{E}_1], \dots, [\mathfrak{D}_i, \mathfrak{E}_k])$$

La proposition 3.4 page 74 implique que $\mathfrak{D}_i \perp \mathfrak{E}$ si et seulement si c' est de poids 1 et pour tout j , $\mathfrak{D}_i \perp \mathfrak{E}_j$. ■

Un corollaire immédiat est la proposition suivante.

Proposition 5.2 – Projections semi-simples et comportements

Un dessein immédiatement total est dans un comportement si et seulement si ses projections semi-simples sont dedans.

La proposition suivante montre que le générateur minimal par combinaison linéaire est l'ensemble des desseins semi-simples.

Proposition 5.3 – Combinaisons linéaires

Pour tout comportement G , on a

$$\text{Cl}(\text{Pss}(G)) = G$$

De plus, pour toute éthique E ,

$$\text{Cl}(E) = G \Leftrightarrow \text{Pss}(G) \subseteq E \subseteq G$$

Démonstration : Pour toute éthique E et tout comportement G , on a $E \subseteq \text{Cl}(E)$, $\text{Pss}(G) \subseteq G$ et $\text{Pss}(E) \subseteq E$.

1. Un dessein immédiatement total (son coefficient à la base est de poids 1) est inclus dans les combinaisons linéaires de coefficient de poids 1 de ses projections semi-simples.

Or les projections semi-simples des desseins de G sont des desseins semi-simples de G (proposition 5.2).

Donc

$$\text{Cl}(\text{Pss}(G)) \supseteq G$$

Or, par monotonie de Cl ,

$$\text{Cl}(\text{Pss}(G)) \subseteq G$$

Donc

$$\text{Cl}(\text{Pss}(G)) = G$$

2. \Rightarrow Supposons $\text{Cl}(E) = G$. On a $E \subseteq G$. De plus, $\text{Pss}(G) = \text{Pss}(\text{Cl}(E)) \subseteq E$.
 \Leftarrow Supposons $\text{Pss}(G) \subseteq E \subseteq G$. D'après le premier cas, $\text{Cl}(\text{Pss}(G)) = G$. On conclut en invoquant la monotonie de Cl . ■

5.1.3 Incarnation

L'incarnation d'un dessein \mathfrak{D} dans un comportement G est la partie de \mathfrak{D} qui est visitée par bi-orthogonal. De manière plus calculatoire, c'est le plus petit dessein du comportement G inclus dans \mathfrak{D} . La notion d'incarnation repose sur le théorème de stabilité (théorème 3.25 page 110). L'inclusion utile ici est l'inclusion pleine (définition 3.44 page 101).

Proposition 5.4 – Action visitée et coefficient de poids 1

Soit \mathfrak{D} et \mathfrak{E} deux desseins orthogonaux et σ une action de \mathfrak{D} visitée lors de l'interaction avec \mathfrak{E} .

Le coefficient préfixant σ est de poids égal à 1.

Démonstration : Soit c le coefficient préfixant σ . L'action σ est visitée donc c apparaît dans $[\mathcal{D}, \mathcal{E}]$. Or $[\mathcal{D}, \mathcal{E}]$ est de poids 1 donc (proposition 3.4 page 74) c est de poids 1. ■

Deux termes sont précisés pour la proposition suivante :

- Annuler un coefficient positif c dans un dessein, c'est le remplacer par 0 et oublier sa continuation. Si ce coefficient suit une action négative alors l'action négative est aussi enlevée (ainsi que sa continuation).
- Un coefficient positif est **visité** si
 - ou bien le coefficient est à la racine du dessein ;
 - ou bien l'action négative précédente est visitée.

Proposition 5.5 – Incarnation

Pour tout comportement G , et tout dessein \mathcal{D} de G , il existe un unique dessein $\mathcal{D}_0 \subseteq \mathcal{D}$ de G tel que

$$\forall \mathcal{E} \in G, \mathcal{E} \subseteq \mathcal{D}_0 \Rightarrow \mathcal{E} = \mathcal{D}_0$$

De plus, ce dessein \mathcal{D}_0 se calcule en annulant les coefficients positifs de \mathcal{D} non visités par un dessein de G^\perp .

Démonstration : On montre l'existence puis l'unicité.

- Existence. Soit \mathcal{D}_0 le dessein obtenu en annulant les coefficients positifs non visités par un dessein de G^\perp .
- Le dessein \mathcal{D}_0 appartient bien à G car il a les mêmes interactions que \mathcal{D} avec les desseins de G^\perp .
- Soit $\mathcal{E} \subseteq \mathcal{D}_0$ un dessein de G .
Supposons que le coefficient positif c' dans \mathcal{E} soit différent du coefficient c correspondant dans \mathcal{D}_0 . Par définition de l'inclusion, c' est obtenu à partir de c en annulant des coefficients non nuls donc son poids est strictement inférieur à 1. Or c est non nul donc visité donc c' l'est aussi. Cela contredit la proposition précédente.
- Unicité. Soit $\mathcal{D}_1 \in G$ un dessein possédant la propriété :

$$\forall \mathcal{E} \in G, \mathcal{E} \subseteq \mathcal{D}_1 \Rightarrow \mathcal{E} = \mathcal{D}_1$$

Soit c un coefficient positif de \mathcal{D}_1 .

- Cas 1. Le coefficient correspondant c' est visité dans \mathcal{D} . Par inclusion des desseins, le coefficient c de \mathcal{D}_1 est inclus dans c' . Supposons que le coefficient c de \mathcal{D}_1 soit strictement inclus dans c' . Son poids est donc strictement inférieur à 1 et, d'après la proposition précédente, il y a contradiction.
- Cas 2. Le coefficient c' correspondant dans \mathcal{D} n'est pas visité. On prouve aisément que c est nul.

On en déduit que $\mathcal{D}_1 = \mathcal{D}_0$. ■

Définition 5.7 – Incarnation

Dans un comportement G , l'**incarnation** $|\mathcal{D}|_G$ d'un dessein \mathcal{D} est le dessein $\mathcal{D}_0 \subseteq \mathcal{D}$ de G tel que

$$\forall \mathcal{E} \in G, \mathcal{E} \subseteq \mathcal{D}_0 \Rightarrow \mathcal{E} = \mathcal{D}_0$$

Proposition 5.6 – Incarnation et dessein positivement complet

Les incarnations des desseins d'un comportement sont des desseins positivement complets.

Démonstration : Soit \mathfrak{D} un dessein matériel (égal à son incarnation) dans un comportement G . D'après la proposition 5.5 page précédente, les coefficients positifs non visités sont annulés (et donc enlevés de l'incarnation). Donc, toutes les actions positives de \mathfrak{D} sont visitées par un dessein de G^\perp et les coefficients préfixant ces actions sont de poids 1 (proposition 5.4 page 132).

Donc le dessein \mathfrak{D} est positivement complet. ■

Proposition 5.7 – Incarnation et intersection

Dans un comportement G , l'incarnation $|\mathfrak{D}|_G$ d'un dessein \mathfrak{D} est caractérisée par $|\mathfrak{D}|_G = \bigcap \{\mathfrak{E} \in G \mid \mathfrak{E} \subseteq \mathfrak{D}\}$

Démonstration : On utilise le théorème de stabilité. ■

Remarque 5.1 *L'intersection en question dans la proposition précédente est l'intersection pleine.*

La notion d'éthique complète permet de définir la complétude interne (voir définition 1.42 page 38) : les connecteurs ne renvoient pas forcément naturellement des comportements mais plutôt des éthiques complètes. En ludique probabiliste, elle est maintenant faite modulo combinaisons linéaires ce qui signifie que seule la partie semi-simple d'une éthique compte.

Définition 5.8 – Éthique complète

Une éthique E est **complète** si

$$\text{Pss}(|E^{\perp\perp}|) \subseteq E$$

L'utilisation de la partie semi-simple $\text{Pss}(\)$ est nécessaire pour le *tenseur* (définition 5.19 page 147) par exemple qui nécessite une clôture par combinaisons linéaires pour obtenir un comportement. C'est naturellement également vrai pour les *décalages* (définition 5.18 page 145), le *plus...*

Proposition 5.8 – Comportement à partir d'une éthique

Tout comportement est la clôture par l'ordre stable de l'ensemble des combinaisons linéaires de ses éthiques complètes.

5.1.4 Superposition

Les combinaisons linéaires interviennent à la racine des desseins mais une autre opération fait intervenir les coefficients pour construire les desseins : la superposition. La superposition relie des desseins simples et des desseins probabilistes.

Dans cette section et par la suite, la notation $E^{\perp p}$ désigne l'orthogonal dans la ludique probabiliste d'une éthique simple et $E^{\perp s}$ son orthogonal dans la ludique simple.

Définition 5.9 – Les fonctions Simp et Proba

Lorsque G est un comportement probabiliste, on note $\text{Simp}(G)$ l'ensemble de ses desseins simples.

Lorsque G est un comportement simple, on note $\text{Proba}(G)$ son bi-orthogonal probabiliste.

Définition 5.10 – Superposition de desseins simples

On dit qu'un dessein probabiliste \mathcal{D} est dans la **superposition** $\text{Sp}(E)$ d'une éthique simple E si

$$\mathcal{D} \in E^{\perp p \perp p}$$

Remarque 5.2 La fonction $\text{Sp}(\)$ généralise la fonction $\text{Proba}(\)$ des comportements simples aux éthiques simples.

Proposition 5.9 – La fonction Proba est injective

La fonction Proba est une injection des comportements simples dans les comportements probabilistes. Son inverse à gauche est Simp .

Démonstration : Soit G un comportement simple. Montrons que $G = \text{Simp}(\text{Proba}(G))$.

- Par monotonie de Simp et Proba , on a $G \subseteq \text{Simp}(\text{Proba}(G))$.
- Montrons que $G \supseteq \text{Simp}(\text{Proba}(G))$.

Soit \mathcal{D} un dessein simple dans $\text{Proba}(G)$. Soit $\mathcal{E} \in G^{\perp s}$. Montrons que $\mathcal{D} \perp \mathcal{E}$.

On a $\mathcal{E} \in G^{\perp s}$ donc $\mathcal{E} \in G^{\perp p}$. Donc $\mathcal{D} \perp \mathcal{E}$. Donc $\mathcal{D} \in G$.

Donc $G \supseteq \text{Simp}(\text{Proba}(G))$ et $G = \text{Simp}(\text{Proba}(G))$. ■

Remarque 5.3 La surjectivité de la fonction Proba n'est pas claire et est laissée en conjecture. De même, une détermination de la notion de superposition sans bi-orthogonal devrait être menée...

Le nom de superposition est utilisé pour évoquer un "mélange" de desseins simples. C'est assez compliqué. En particulier, la remarque 5.6 page 140 indique qu'on ne peut se contenter de regarder les projections simples.

La notion suivante permet de se passer de la surjectivité de la fonction Proba par la suite.

Définition 5.11 – Comportement s-régulier

Un comportement probabiliste G est **s-régulier** pour un comportement simple H si

- $H = \text{Simp}(G)$
- $H^{\perp s} = \text{Simp}(G^{\perp p})$

Un comportement probabiliste est **s-régulier** s'il l'est pour un comportement simple donné.

Proposition 5.10 – Caractérisation des comportements s-réguliers

Un comportement probabiliste est s-régulier si et seulement si

$$\text{Simp}(G^{\perp p}) = \text{Simp}(G)^{\perp s}$$

Proposition 5.11 – Existence de comportements s-réguliers

Soit H un comportement simple.

Il existe un comportement probabiliste G s-régulier pour H .

Démonstration : Soit $G = \text{Proba}(H)$. Montrons que

- $H = \text{Simp}(G)$
- $H^{\perp s} = \text{Simp}(G^{\perp p})$

Le premier point est impliqué par la proposition 5.9 page précédente.

On a $H^{\perp s} \perp H$ donc $H^{\perp s} \subseteq H^{\perp p}$ donc $H^{\perp s} \perp G$.

Donc $H^{\perp s} \subseteq G^{\perp}$, ce qui donne $H^{\perp s} \subseteq \text{Simp}(G^{\perp})$

Or $H \subseteq G$ donc $H \perp \text{Simp}(G^{\perp p})$ donc $\text{Simp}(G^{\perp p}) \subseteq (\text{Simp}(G))^{\perp s}$ et

$$\text{Simp}(G^{\perp p}) = (\text{Simp}(G))^{\perp s} = H^{\perp s}$$

■

Remarque 5.4 L'orthogonal d'un comportement s-régulier est s-régulier.

Définition 5.12 – Connecteur s-régulier

Un connecteur est **s-régulier** s'il envoie des comportements s-réguliers sur des comportements s-réguliers.

Proposition 5.12 – Dual d'un connecteur s-régulier

Le dual d'un connecteur s-régulier est s-régulier.

Proposition 5.13 – Connecteurs habituels et s-régularité

Les connecteurs habituels sont s-réguliers.

Démonstration :

- On le vérifie simplement pour le *avec*.

$$\begin{aligned} \text{Simp}((G_1 \& G_2)^{\perp p})^{\perp s} &= \text{Simp}(G_1^{\perp p} \oplus G_2^{\perp p})^{\perp s} \\ &= \text{Simp}(G_1^{\perp p})^{\perp s} \& \text{Simp}(G_2^{\perp p})^{\perp s} \\ &= \text{Simp}(G_1) \& \text{Simp}(G_2) \end{aligned}$$

- Par passage au dual, le connecteur *plus* est donc aussi s-régulier.
- On procède de manière similaire pour les autres connecteurs.

■

Proposition 5.14 – Incarnation et desseins simples

Pour tout comportement probabiliste G ,

$$\text{Simp}(|G|) = |\text{Simp}(G)|$$

Démonstration : On montre la double inclusion.

– Soit $\mathcal{D} \in \text{Simp}(|G|)$.

On a

$$\mathcal{D} = |\mathcal{D}|_G = \cap\{\mathcal{D}' \in G | \mathcal{D}' \subseteq \mathcal{D}\} = \cap\{\mathcal{D}' \in \text{Simp}(G) | \mathcal{D}' \subseteq \mathcal{D}\} = |\mathcal{D}|_{\text{Simp}(G)}$$

Donc $\mathcal{D} \in |\text{Simp}(G)|$.

– Soit $\mathcal{D} \in |\text{Simp}(G)|$.

On a

$$\mathcal{D} = |\mathcal{D}|_{\text{Simp}(G)} = \cap\{\mathcal{D}' \in \text{Simp}(G) | \mathcal{D}' \subseteq \mathcal{D}\} = \cap\{\mathcal{D}' \in G | \mathcal{D}' \subseteq \mathcal{D}\} = |\mathcal{D}|_G$$

Donc $\mathcal{D} \in \text{Simp}(|G|)$. ■

La suite de la section explique une partie de la difficulté de la caractérisation de la superposition en général et explicite le cas des desseins finis.

La notion suivante provient de la remarque qu'un coefficient de poids différent de 1 dans un dessein n'est pas visité par un dessein orthogonal (proposition 5.4 page 132).

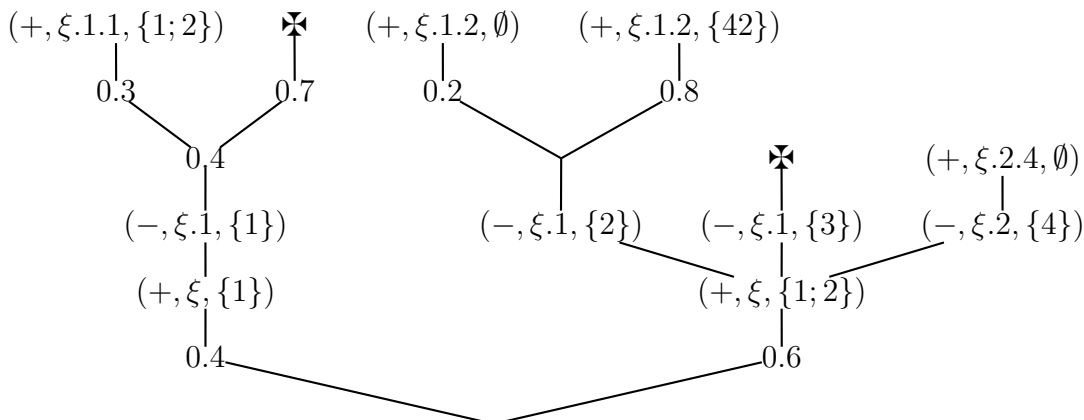
Définition 5.13 – Projection simple complète

Une **projection simple complète** d'un dessein probabiliste est une projection simple privée des actions propres (et de leur continuation) dont les coefficients préfixes sont de poids différents de 1 ainsi que des actions négatives propres maximales une fois la première opération achevée.

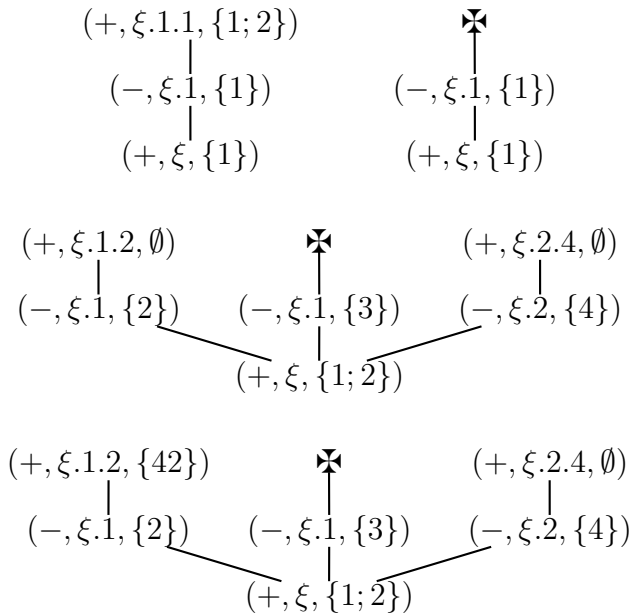
Remarque 5.5 Un dessein négatif $c.(\mathcal{D}_1, \dots, \mathcal{D}_n)$ tel que $\mathcal{W}(c) < 1$ a une unique projection simple complète : la divergence négative \mathfrak{Fid}^- .

Un dessein positif $c.(\mathcal{D}_1, \dots, \mathcal{D}_n)$ tel que $\mathcal{W}(c) < 1$ a une unique projection simple complète : la divergence positive \mathfrak{Fid} .

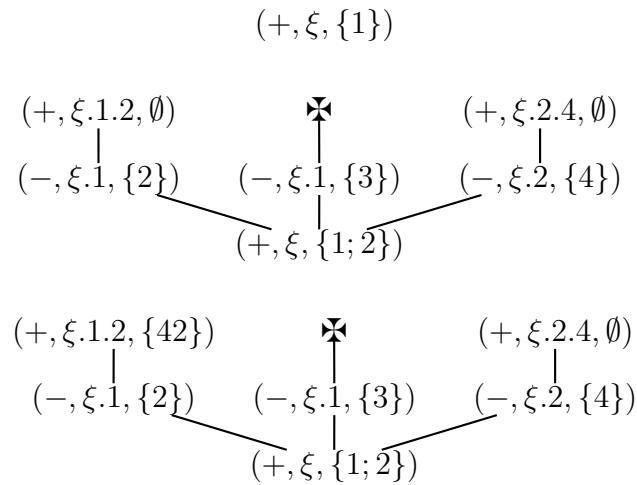
Ainsi, les projections simples de



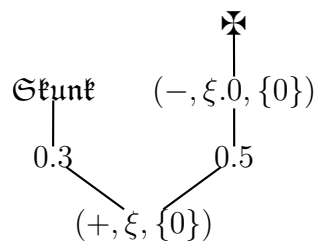
sont



et ses projections simples complètes sont :



Les projections simples complètes de



sont réduites à

$$\begin{array}{c} \mathfrak{Fid}^- \\ | \\ (+, \xi, \{0\}) \end{array}$$

alors que ses projections simples sont

$$\begin{array}{ccc} \mathfrak{Fid}^- & & \mathfrak{X} \\ | & & | \\ (+, \xi, \{0\}) & (+, \xi, \{0\}) & (-, \xi, 0, \{0\}) \\ & & | \\ & & (+, \xi, \{0\}) \end{array}$$

Proposition 5.15 – Projections simples de desseins de profondeur finie

1. Soit \mathcal{D} est un dessein de profondeur finie. Le dessein \mathcal{D} appartient à un comportement probabiliste G si et seulement si toutes ses projections simples complètes appartiennent à G .
2. De plus, si $\mathcal{D} \in G$, les projections simples de \mathcal{D} sont dans G .

Démonstration :

1. On fait une induction sur la hauteur h de \mathcal{D} définie ainsi :

$$h : \begin{cases} \mathbf{0} & \mapsto 0 \\ \mathfrak{X} & \mapsto 0 \\ \mathfrak{X}_\xi & \mapsto 0 \\ (s_1.\mathcal{D}_1, \dots, s_n.\mathcal{D}_n) & \mapsto 1 + \max h(\mathcal{D}_i) \\ (+, \xi, I).(\mathcal{D}_1, \dots, \mathcal{D}_n) & \mapsto 1 + \max h(\mathcal{D}_i) \\ ((-, \xi, I).\mathcal{D}_I) & \mapsto 1 + \sup h(\mathcal{D}_I) \end{cases}$$

- Si $h = 0$, alors \mathcal{D} est un démon ou $\mathbf{0}$ donc \mathcal{D} est son unique projection simple (en identifiant $\mathbf{0}$ à \mathfrak{Fid} ou \mathfrak{Fid}^- suivant la polarité).
- Si $h = n + 1$ et pour tout dessein \mathcal{D}' de hauteur n , et tout comportement H , \mathcal{D}' appartient à H si et seulement si ses projections simples complètes sont dans H . On procède par cas sur la forme de \mathcal{D} .
 - Cas 1. $\mathcal{D} = (s_1.\mathcal{D}_1, \dots, s_n.\mathcal{D}_n)$.
Si $\sum_i s_i < 1$ alors \mathcal{D} n'appartient à aucun comportement. Or, ses projections simples complètes sont réduites à la divergence (positive ou négative suivant la polarité) qui n'appartiennent bien à aucun comportement.
On suppose donc par la suite que $\sum_i s_i = 1$.
 - Si \mathcal{D} est positif. On a pour tout dessein $\mathfrak{E} \in G^\perp$:

$$\llbracket \mathcal{D}, \mathfrak{E} \rrbracket = (s_1. \llbracket \mathcal{D}_1, \mathfrak{E} \rrbracket, \dots, s_n. \llbracket \mathcal{D}_n, \mathfrak{E} \rrbracket)$$

Supposons $\mathcal{D} \in G$. On a $\mathcal{W}(\llbracket \mathcal{D}, \mathfrak{E} \rrbracket) = 1$. D'après la proposition 3.4 page 74, on déduit que pour tout i , $\mathcal{W}(\llbracket \mathcal{D}_i, \mathfrak{E} \rrbracket) = 1$. Donc, par induction, les projections simples complètes de \mathcal{D}_i sont dans G . On conclut en remarquant que les projections simples complètes des \mathcal{D}_i sont exactement les projections simples complètes de \mathcal{D} .

Réciproquement, si $\mathcal{D} \notin G$, il existe un \mathfrak{E} tel que $\mathcal{W}(\llbracket \mathcal{D}, \mathfrak{E} \rrbracket) < 1$. Pour un tel \mathfrak{E} , il existe un i tel que $\mathcal{W}(\llbracket \mathcal{D}_i, \mathfrak{E} \rrbracket) < 1$. Par induction, on obtient une projection simple complète de \mathcal{D} qui n'est pas dans G .

– Si \mathcal{D} est négatif.

Le dessein \mathfrak{E} est de la forme $c.(\mathfrak{E}_1, \dots, \mathfrak{E}_k)$ où les (\mathfrak{E}_i) sont semi-simples. On a

$$\llbracket \mathcal{D}, \mathfrak{E} \rrbracket = c.(c_1, \dots, c_n)$$

où, pour tout $i \in \{1, \dots, k\}$,

$$c_i = (s_1. \llbracket \mathcal{D}_1, \mathfrak{E}_i \rrbracket, \dots, s_n. \llbracket \mathcal{D}_n, \mathfrak{E}_i \rrbracket)$$

On conclut en procédant comme pour le cas précédent (\mathcal{D} positif).

– Cas 2. $\mathcal{D} = (+, \xi, I).(\mathcal{D}'_1, \dots, \mathcal{D}'_n)$.

Supposons $\mathcal{D} \in G$. Soit $\mathfrak{E} \in G^\perp$. Le dessein \mathfrak{E} est de la forme $c.(\mathfrak{E}_1, \dots, \mathfrak{E}_k)$ où les (\mathfrak{E}_i) sont semi-simples.

Chaque \mathfrak{E}_i s'écrit

$$\mathfrak{E}_i = \dots (-, \xi, I). \mathfrak{E}_I^{(i)} \dots$$

Et pour tout i , $\mathcal{W}(\llbracket \mathfrak{E}_I^{(i)}, \mathcal{D}_i \rrbracket) = 1$.

L'induction s'applique aux \mathcal{D}_i comme précédemment dans les comportements adéquats.

Réciproquement, le cas $\mathcal{D} \notin G$ se traite de manière similaire.

– Cas 3. $\mathcal{D} = ((-, \xi, I). \mathcal{D}'_I)_I$. On procède de même.

2. Ce cas se déduit du cas 1 car les comportements sont clos par ordre extensionnel. ■

Proposition 5.16 – Actions visitées

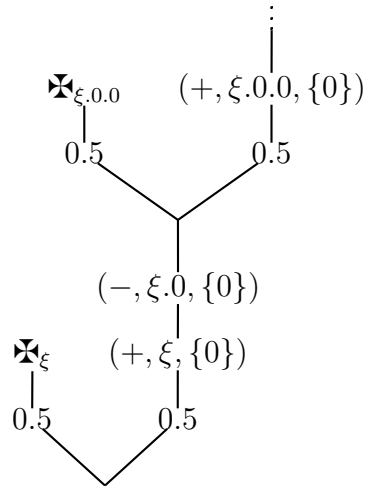
Soit G un comportement simple. Les desseins de G et ceux de $G^{\perp_p \perp_p}$ visitent les mêmes actions sur les desseins de G^{\perp_s} .

Démonstration : Le comportement G est inclus dans $G^{\perp_p \perp_p}$ donc les actions visitées sur les desseins de G^{\perp_s} par les desseins de G sont incluses dans les actions visitées par les desseins de $G^{\perp_p \perp_p}$. La réciproque se prouve en passant par les desseins de profondeur finie.

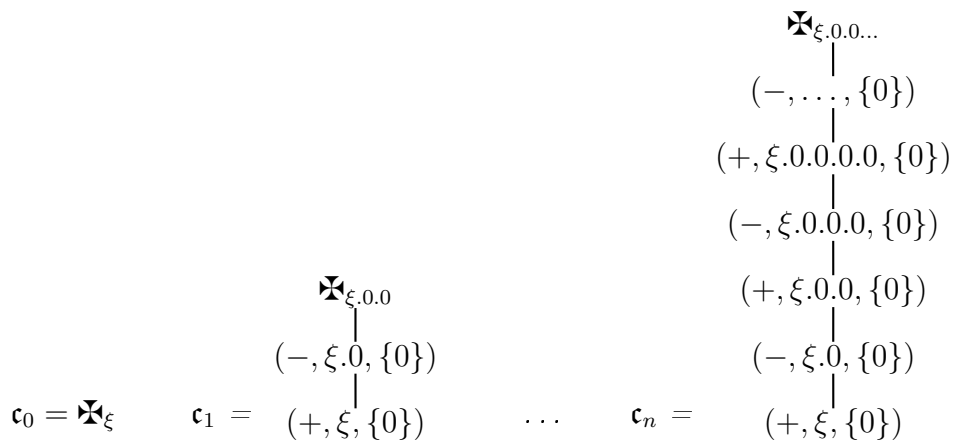
Soit σ (l'occurrence d') une action d'un dessein $\mathfrak{E} \in G^{\perp_s}$ visitée par un dessein $\mathcal{D} \in G^{\perp_p \perp_p}$. Cette action est visitée en temps fini, c'est-à-dire qu'un sous-dessein fini \mathcal{D}_0 de \mathcal{D} visite aussi cette action. Soit \mathcal{D}' un dessein contenant \mathcal{D}_0 supérieur à \mathcal{D} pour l'ordre extensionnel (une construction possible est de rajouter toutes les actions négatives possibles dans \mathcal{D}_0 et de mettre des démons après). On a obtenu un dessein \mathcal{D}' de profondeur finie visitant σ . Par la proposition 5.15 page précédente, on déduit que les projections simples complètes de \mathcal{D}' sont dans $G^{\perp_p \perp_p}$. De plus, puisque \mathcal{D}' visite σ , une de ses projections simples visite σ . Donc il existe un dessein simple de $G^{\perp_p \perp_p}$ qui visite σ . Or $G^{\perp_p \perp_p}$ et G ont les mêmes desseins simples (proposition 5.9 page 135) donc il existe un dessein simple de G qui visite σ . ■

Remarque 5.6 *L'hypothèse de profondeur finie est indispensable.*

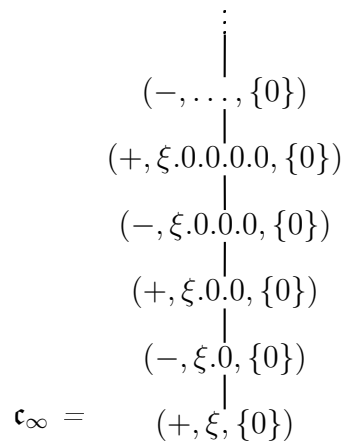
En effet, le dessein



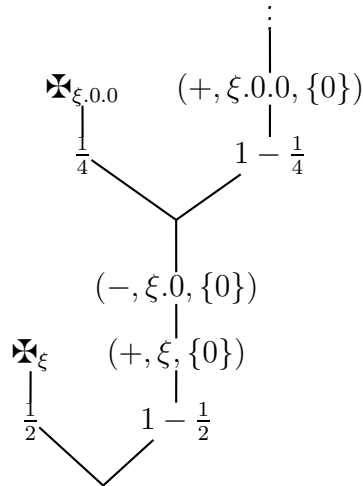
est dans la superposition de l'ensemble des chroniques finies de la forme



même s'il est aussi dans la superposition de l'ensemble constitué des c_n et de la chronique infinie :



Par contre, le dessein (inspiré de l'exemple page 73) :



appartient à la superposition de $\{c_i \mid i \in \mathbb{N} \cup \{\infty\}\}$ mais pas à celle de $\{c_i \mid i \in \mathbb{N}\}$ car le poids de la chronique infinie est strictement positif. Cependant, ce phénomène n'est pas la seule difficulté posée par la superposition.

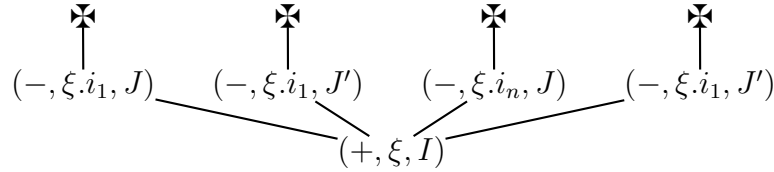
5.1.5 Notions pour la complétude interne des comportements

Pour donner une éthique complète aux connecteurs multiplicatifs, on reprend la notion d'indépendance de la ludique simple (section 1.5.3 page 41).

On définit la notion de répertoire d'un comportement de manière similaire à la ludique simple (définition 1.46 page 40).

Définition 5.14 – Répertoire d'un comportement

– Le **répertoire** d'un comportement positif G est l'ensemble des ramifications I telles que le dessein suivant appartienne à G



où $\begin{cases} I = \{i_1, \dots, i_n\} \\ J, J' \text{ parcourent } \mathcal{P}_{fin}(\mathbb{N}) \end{cases}$

– Le **répertoire** d'un comportement négatif est l'ensemble des ramifications I

telles que le dessein $(-\, \xi, I)$ est inclus dans l'incarnation dans G de

$\mathfrak{D}ai^- = (-, \xi, J) \dots (-, \xi, J')$ où J, J' parcourent $\mathcal{P}_{fin}(\mathbb{N})$.

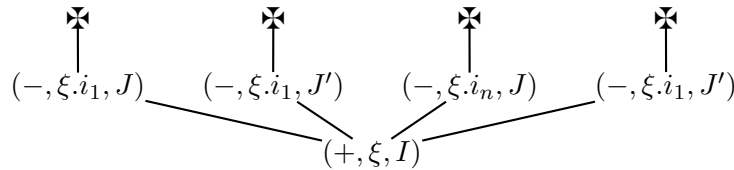
– le **répertoire** d'une éthique est le répertoire de son bi-orthogonal.

Proposition 5.17 – Caractérisation du répertoire d'un comportement positif

Le répertoire d'un comportement positif G sur la base $\vdash \xi$ est l'ensemble des ramifications I tel qu'un dessein semi-simple de G commence par $(+, \xi, I)$.

Démonstration : Par monotonie (proposition 3.27 page 111), le répertoire d'un comportement positif sur la base $\vdash \xi$ contient l'ensemble des ramifications I tel qu'un dessein semi-simple de G commence par $(+, \xi, I)$.

La réciproque est évidente : si le dessein



appartient à G alors il y a bien un dessein semi-simple de G qui commence par $(+, \xi, I)$. ■

Proposition 5.18 – Répertoire

Tout comportement sur une base unaire a le même répertoire que son orthogonal.

Démonstration : Soit G un comportement positif sur la base $\vdash \xi$. Soit I une ramification de son répertoire et \mathfrak{D} un dessein commençant par $(+, \xi, I)$ dans G . Le réseau $\mathfrak{D}, |\mathfrak{D}ai^-|_{G^\perp}$ converge donc I appartient au répertoire de G^\perp .

Réciproquement, si I est dans le répertoire de G^\perp alors $(-, \xi, I)$ est visité dans $\mathfrak{D}ai^-$ par un dessein de G et donc (par la proposition 5.2 page 132) il y a un dessein semi-simple dans G qui commence par $(+, \xi, I)$. On en déduit que I appartient au répertoire de G . ■

Définition 5.15 – Comportements disjoints et connexes

- Deux comportements sont **disjoints** si leurs répertoires sont disjoints.
- Une éthique E est **connexe** si son répertoire est un singleton $\{I\}$. L'ensemble I est appelé la **ramification** de E .

Proposition 5.19 – Caractérisation des comportements disjoints

- Deux comportements positifs sont disjoints si et seulement si leur intersection est réduite aux combinaisons linéaires de poids 1 de démons (focalisés ou non).
- Deux comportements négatifs sont disjoints si et seulement si

$$\forall \mathfrak{D} \in \text{Pss}(G), \forall \mathfrak{E} \in \text{Pss}(H), |\mathfrak{D}|_G \cap |\mathfrak{E}|_H = \emptyset.$$

Démonstration : On montre les deux implications dans le cas positif.

\Rightarrow Soient G et H deux comportements disjoints. Soit $\mathfrak{D} \in G \cap H$. Soit \mathfrak{D}' une projection semi-simple de \mathfrak{D} . On a $\mathfrak{D}' \in G \cap H$. D'après la proposition 5.17 page précédente, \mathfrak{D}' ne commence pas par une action propre. C'est donc un démon (focalisé ou non). On en déduit que toutes les projections semi-simples de \mathfrak{D} sont des démons et donc que \mathfrak{D} est une combinaison linéaire de démons.

\Leftarrow On utilise la proposition 5.17 page précédente. ■

Comme pour leurs équivalents en ludique simple, les propositions suivantes permettent de prouver la complétude interne pour le *tenseur* (proposition 5.26 page 148).

Proposition 5.20 – Existence d'une projection

Soit $\mathbb{M} \subseteq \mathbb{N}$ un réservoir. Tout dessein positif semi-simple $\mathfrak{D} = (+, \xi, I)$. \mathfrak{R} est égal à $\mathfrak{D}_{\mathbb{M}} \otimes \mathfrak{D}_{\mathbb{N} \setminus \mathbb{M}}$ où $\mathfrak{D}_{\mathbb{M}}$ commence par $(+, \xi, I \cap \mathbb{M})$ et $\mathfrak{D}_{\mathbb{N} \setminus \mathbb{M}}$ commence par $(+, \xi, I \setminus \mathbb{M})$. Le dessein $\mathfrak{D}_{\mathbb{M}}$ est le **projeté** de \mathfrak{D} sur \mathbb{M} .
On étend cette définition en posant $\mathfrak{X}_{\xi_{\mathbb{M}}} = \mathfrak{X}_{\xi}$.

Démonstration : On reprend la preuve de la proposition 1.15 page 43. ■

Proposition 5.21 – Projection

Soit E une éthique connexe et \mathbb{M} un réservoir. On a $(E^{\perp\perp})_{\mathbb{M}} = (E_{\mathbb{M}})^{\perp\perp}$.

Démonstration : La preuve est similaire à celle dans le cadre de la ludique simple [Gir01] pour les desseins semi-simples. La preuve complète est alors immédiate en utilisant les combinaisons linéaires. ■

Définition 5.16 – Indépendance

Deux répertoires R_1 et R_2 sont **indépendants** si

$$\forall I_1, J_1 \in R_1, I_2, J_2 \in R_2, I_1 \cup I_2 = J_1 \cup J_2 \Rightarrow I_1 = J_1, I_2 = J_2$$

Deux comportements sont **indépendants** si leurs répertoires sont indépendants.

5.2 Connecteurs

Les connecteurs de la ludique probabiliste généralisent ceux de la ludique simple (section 1.5 page 38). La notion de complétude interne (le respect des éthiques complètes) doit être adaptée (définition 5.8 page 134).

5.2.1 Décalages

Les décalages de desseins (définition 5.17) doivent être définis spécifiquement pour le cadre probabiliste car ils interagissent avec les coefficients à la racine des desseins. Ils sont à rapprocher de la définition 2.13 page 59.

Définition 5.17 – Décalage d'un dessein

– Si \mathcal{D} est un dessein positif de base $\vdash \xi.i$ alors

$$\uparrow \mathcal{D} = \begin{cases} (-, \xi, \{i\}).\mathcal{D} & \text{si } \mathcal{D} \neq \mathfrak{F}id \\ \mathfrak{G}\mathfrak{k}\mathfrak{u}\mathfrak{n}\mathfrak{k} & \text{sinon} \end{cases}$$

– Si \mathcal{D} est un dessein négatif de base $\xi.i \vdash$ alors

$$\downarrow \mathcal{D} = (+, \xi, \{i\}).\mathcal{D}$$

Définition 5.18 – Décalages

Sur une base unaire, le décalage d'un comportement est le bi-orthogonal des décalages de ses desseins. Lorsque G est positif de base $\vdash \xi.i$ (resp. négatif de base $\xi.i \vdash$), on le note $\uparrow G$ (resp. $\downarrow G$).

Proposition 5.22 – Complétude interne pour les décalages

Soit G un comportement.

– Si G est positif, alors $\{\uparrow \mathcal{D} \mid \mathcal{D} \in G\}$ est une éthique complète pour $\uparrow G$.

– Si G est négatif de base $\xi.i \vdash$, alors $\{\downarrow \mathcal{D} \mid \mathcal{D} \in G\} \cup \{\mathfrak{X}_\xi\}$ est une éthique complète pour $\downarrow G$.

Démonstration : Pour les desseins semi-simples, on fait la même preuve qu'en ludique simple. On conclut avec la proposition 5.3 page 132. ■

5.2.2 Additifs

Les connecteurs additifs sont définis exactement de la même manière que dans le cadre de la ludique simple 1.5.2 page 39.

La complétude interne pour le *avec* est simplement le mystère de l'incarnation (voir proposition 1.12 page 41).

Proposition 5.23 – Complétude interne pour le avec

Pour tous comportements disjoints G et H sur une même base négative,

$$\text{Pss}(|G \& H|) = \text{Pss}(|G|) \times \text{Pss}(|H|)$$

Démonstration : La preuve est similaire à celle en ludique simple : un dessin matériel (qui est sa propre incarnation) semi-simple de $G \& H$ est simplement l'*union (par hypothèse disjointe)* d'un dessin matériel semi-simple de G et d'un dessin matériel semi-simple de H . ■

La complétude interne pour le *plus* est obtenue par la propriété de la disjonction (proposition 1.13 page 41).

Proposition 5.24 – Complétude interne pour le plus

Pour tous comportements positifs disjoints,

$$G \oplus H = \text{Cl}(G \cup H)$$

Autrement dit,

$$\text{Pss}(G \oplus H) = \text{Pss}(G) \cup \text{Pss}(H)$$

Démonstration : Similaire au cadre de la ludique simple. ■

Remarque 5.7 *On ne peut pas généraliser ces propositions à tous les desseins car une combinaison linéaire de desseins de G et de H mélangés n'est pas forcément une combinaison linéaire de desseins uniquement de G ou une combinaison linéaire de desseins uniquement de H .*

5.2.3 Multiplicatifs

Contrairement aux connecteurs additifs qui sont définis aux niveau des comportements, les différentes versions du *tenseur* font interagir deux desseins entre eux. Les différentes définitions du *tenseur* doivent donc être adaptées au cadre des coefficients.

Cela se passe simplement en utilisant la *bilinéarité* du tenseur.

Définition 5.19 – Le tenseur

Les différentes versions du tenseur sont définies par bilinéarité :

- Pour des desseins semi-simples, la construction est similaire à la construction en ludique simple (définitions 1.52 page 42 et 2.12 page 58).
- Le produit tensoriel de deux comportements est l'ensemble des combinaisons linéaires des produits tensoriels des parties semi-simples

$$\text{Pss}(G \otimes H) = \text{Pss}(G) \otimes \text{Pss}(H)$$

et donc

$$G \otimes H = \text{Cl}(\text{Pss}(G) \otimes \text{Pss}(H))$$

On pourrait aussi définir un *tenseur* entre desseins arbitraires (non forcément semi-simples) de la manière suivante :

$$= c.(\mathcal{D}_1, \dots, \mathcal{D}_n) \otimes c'.(\mathcal{D}'_1, \dots, \mathcal{D}'_m) \\ = c.(c'.(\mathcal{D}_1 \otimes \mathcal{D}'_1, \dots, \mathcal{D}_1 \otimes \mathcal{D}'_m), \dots, c'.(\mathcal{D}_n \otimes \mathcal{D}'_1, \dots, \mathcal{D}_n \otimes \mathcal{D}'_m))$$

ou

$$= c'.(\mathcal{D}_1, \dots, \mathcal{D}_n) \otimes c.(\mathcal{D}'_1, \dots, \mathcal{D}'_m) \\ = c.(c'.(\mathcal{D}_1 \otimes \mathcal{D}'_1, \dots, \mathcal{D}_1 \otimes \mathcal{D}'_m), \dots, c'.(\mathcal{D}_n \otimes \mathcal{D}'_1, \dots, \mathcal{D}_n \otimes \mathcal{D}'_m))$$

Ces deux choix donneraient lieu à une nouvelle classification de \otimes entre desseins dans l'esprit de la classification tenseur commutatif/non commutatif. Cependant, la différence entre ces deux choix pour le *tenseur* entre desseins non semi-simples n'est pas observable au niveau des comportements et on ne se préoccupe pas plus avant d'une éventuelle classification.

Proposition 5.25 – Adjoints

Des adjoints peuvent encore être définis.

- Le tenseur gauche \otimes tel que

$$c.(\mathcal{D}_1, \dots, \mathcal{D}_n) \otimes c'.(\mathcal{D}'_1, \dots, \mathcal{D}'_m) \\ = c.(c'.(\mathcal{D}_1 \otimes \mathcal{D}'_1, \dots, \mathcal{D}_1 \otimes \mathcal{D}'_m), \dots, c'.(\mathcal{D}_n \otimes \mathcal{D}'_1, \dots, \mathcal{D}_n \otimes \mathcal{D}'_m))$$

a un adjoint

$$[[\mathfrak{F}, \mathcal{D}_1 \otimes \mathcal{D}_2]] = [[\mathfrak{F}(\mathcal{D}_2), \mathcal{D}_1]]$$

- Le tenseur droit \otimes tel que

$$c'.(\mathcal{D}_1, \dots, \mathcal{D}_n) \otimes c.(\mathcal{D}'_1, \dots, \mathcal{D}'_m) \\ = c.(c'.(\mathcal{D}_1 \otimes \mathcal{D}'_1, \dots, \mathcal{D}_1 \otimes \mathcal{D}'_m), \dots, c'.(\mathcal{D}_n \otimes \mathcal{D}'_1, \dots, \mathcal{D}_n \otimes \mathcal{D}'_m))$$

a un adjoint

$$[[\mathfrak{F}, \mathcal{D}_1 \otimes \mathcal{D}_2]] = [[\mathfrak{F}(\mathcal{D}_1), \mathcal{D}_2]]$$

Démonstration : Pour des desseins semi-simples, les adjoints se définissent exactement de la même manière qu'en ludique simple.

Pour des desseins non semi-simples, on utilise les adjoints semi-simples et des coefficients. Soit $\mathfrak{F} = c_{\mathfrak{F}}(\mathfrak{F}^1, \dots, \mathfrak{F}^l)$ un dessein négatif. On pose :

$$\mathfrak{F}(c.(\mathcal{D}_1, \dots, \mathcal{D}_n)) = c(c_{\mathfrak{F}}(\mathfrak{F}^1(\mathcal{D}_1), \dots, \mathfrak{F}^l(\mathcal{D}_1)), \dots, c_{\mathfrak{F}}(\mathfrak{F}^1(\mathcal{D}_n), \dots, \mathfrak{F}^l(\mathcal{D}_n)))$$

où $\mathfrak{F}^i(\mathfrak{D}_j)$ est l'adjoint pour des desseins semi-simples. ■

Les différentes versions du *par* sont simplement définies par dualité.

Proposition 5.26 – Complétude interne pour le tenseur

Si G_1 et G_2 sont positifs et indépendants alors $\{\mathfrak{D}_1 \otimes \mathfrak{D}_2 \mid \mathfrak{D}_1 \in G_1, \mathfrak{D}_2 \in G_2\}$ et $\{\mathfrak{D}_1 \otimes \mathfrak{D}_2 \mid \mathfrak{D}_1 \in \text{Pss}(G_1), \mathfrak{D}_2 \in \text{Pss}(G_2)\}$ sont des éthiques complètes pour $G_1 \otimes G_2$.

Démonstration : Comme pour la ludique simple, on utilise la proposition d'existence de projection (proposition 5.20 page 144). ■

5.2.4 Quantificateurs

Les quantificateurs (vus comme des connecteurs additifs infinis) sont définis de manière analogue (section 1.6 page 43) à la ludique simple. Le quantificateur universel est simplement une intersection de comportements alors que le quantificateur existentiel est une union close par bi-orthogonal.

Chapitre 6

Ludique probabiliste : uniformité

Sommaire

6.1	Distance	150
6.2	Bi-comportements	152
6.3	Connecteurs	157
6.3.1	Délocalisation et décalage	157
6.3.2	Additifs	158
6.3.3	Multiplicatifs	159
6.4	Complétude vis-à-vis de la ludique simple	160
6.4.1	Décalages	162
6.4.2	Additifs	163
6.4.3	Multiplicatifs	164
6.4.4	Les quantificateurs	164
6.4.5	Séquents de bi-comportements	165
6.5	Complétude forte	165
6.5.1	Condition de gain	165
6.5.2	Correction	166
6.5.3	Complétude	167

Le passage de la ludique simple à la ludique probabiliste impose de transformer la notion de bi-comportement avec PER (définition 1.59 page 44) en une version “probabiliste”. La présence de coefficients incite à définir une notion de distance (section 6.1) pour pouvoir parler de desseins proches ou éloignés.

Ainsi, l’uniformité utilise une notion interactive de “distance” (sans contraintes sur les desseins à distance nulle). Les bi-comportements sont des paires composées d’un comportement probabiliste et d’une distance sur les desseins partiels du comportement. Les desseins uniformes sont les desseins à distance nulle. Cette notion d’uniformité permet de définir la notion de gain dans le cadre probabiliste qui donne un résultat de complétude forte (section 6.5 page 165) vis-à-vis de la ludique simple et, par composition, pour $MALL_2$ et $MAAL_2$.

Sauf indication contraire, les bases sont supposées unaires.

6.1 Distance

La notion de distance la plus simple à donner est celle sur les coefficients. Une notion de distance sur les desseins en découle.

Définition 6.1 – Distance sur les coefficients

La **distance** entre deux coefficients finis est définie ainsi :

$$\delta_c(\mathbf{0}, \mathbf{0}) = 0$$

$$\begin{aligned} \delta_c((s_1.c_1, \dots, s_n.c_n), (s'_1.c'_1, \dots, s'_n.c'_n)) \\ = \frac{1}{n} \sum_i (|s_i - s'_i| + \min(s_i, s'_i) \cdot \delta_c(c_i, c'_i)) \end{aligned}$$

$$\delta_c(c, c') = 1 \quad \text{sinon}$$

La distance entre deux coefficients quelconques c et c' est la limite supérieure de la suite des distances entre les approximations supérieures de longueur n de c et c' .

La distance vérifie les propriétés suivantes :

Proposition 6.1 – Distance sur les coefficients

La distance entre deux coefficients est comprise dans l’intervalle $[0; 1]$ et

$$\delta_c(c, c') = 0 \iff c = c'$$

De plus, δ_c est symétrique.

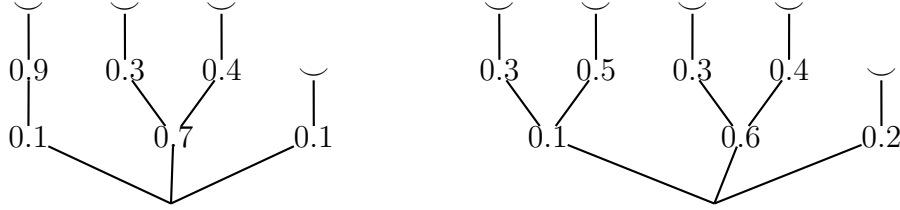
Remarque 6.1 La distance est bornée par 1 grâce au min dans la définition du cas suivant.

$$\delta_c((s_1.c_1, \dots, s_n.c_n), (s'_1.c'_1, \dots, s'_n.c'_n)) = \frac{1}{n} \sum_i (|s_i - s'_i| + \min(s_i, s'_i) \cdot \delta_c(c_i, c'_i))$$

Avec un max, par exemple, la distance pourrait “exploser”.

L'intérêt d'avoir une distance bornée est qu'on peut prendre un sup infini dans la définition de l'orthogonal d'une bi-éthique (définition 6.7 page 154).

Ainsi, la distance entre les coefficients suivants :



$$\text{est } 0.1 = \frac{1}{3} \sum (0 + 0.1 * 1; 0.1 + 0.6 * 0; 0.1 + 0.1 * 0).$$

De la distance sur les coefficients δ_c , on peut déduire une distance δ sur les desseins. Celle-ci a une utilité technique assez limitée mais sert d'inspiration pour la distance qui intervient dans la notion de bi-éthique (définition 6.6 page 153).

Définition 6.2 – Distance sur les desseins

La distance δ_c est étendue aux desseins de la manière suivante :

$$\delta((s_1.\mathcal{D}_1, \dots, s_n.\mathcal{D}_n), (s'_1.\mathcal{D}'_1, \dots, s'_n.\mathcal{D}'_n)) = \frac{1}{n} \sum_i (|s_i - s'_i| + \min(s_i, s'_i) \cdot \delta(\mathcal{D}_i, \mathcal{D}'_i))$$

$$\delta(K^+.\mathcal{D}_1, \dots, \mathcal{D}_n, K^+.\mathcal{D}'_1, \dots, \mathcal{D}'_n) = \max_i \delta(\mathcal{D}_i, \mathcal{D}'_i)$$

$$\delta(\mathbf{X}, \mathbf{X}) = 0$$

$$\delta(\mathbf{X}_\xi, \mathbf{X}_\xi) = 0$$

$$\delta(\mathbf{0}, \mathbf{0}) = 0$$

$$\delta(((-, \xi, I).\mathcal{D}_I)_{I \in \mathcal{P}_{fin}(\mathbb{N})}, ((-, \xi, I).\mathcal{D}'_I)_{I \in \mathcal{P}_{fin}(\mathbb{N})}) = \max_I \delta(\mathcal{D}_I, \mathcal{D}'_I)$$

$$\delta(\mathfrak{F}id^-, \mathfrak{F}id^-) = 0$$

$$\delta(\mathcal{D}, \mathcal{D}') = 1$$

sinon

Proposition 6.2 – La distance δ

La fonction δ est bien une distance.

La notion de distance est relâchée en la notion de pré-distance qui suffit pour les définitions *interactives* de bi-éthique (définition 6.6 page 153). Pour l'uniformité, on ne pourrait pas garder l'équivalence $d(x, y) = 0 \iff x = y$ qui est fausse interactivement pour la définition d'orthogonal considérée (définition 6.7 page 154).

Définition 6.3 – Pré-distance

Une **pré-distance** d sur un ensemble X est une fonction symétrique

$$d : X \times X \rightarrow [0; 1]$$

6.2 Bi-comportements

La notion de dessein partiel est similaire (définition 1.60 page 44) à celle dans le cadre de la ludique simple.

Définition 6.4 – Dessein partiel

Un **dessein partiel d'une éthique** E est un dessein pleinement inclus dans un dessein de cette éthique.

L'ensemble des desseins partiels est noté E^p .

Remarque 6.2 Le dessein négatif $\mathfrak{F}id^- = \mathbf{0}$ n'est un dessein partiel d'aucune éthique.

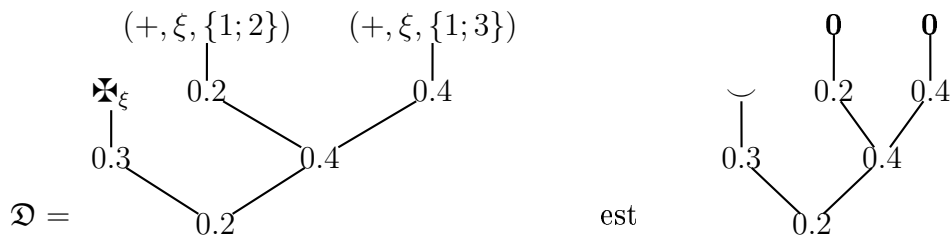
La définition suivante permet de parler de la partie d'un coefficient qui précède un ensemble d'actions.

Définition 6.5 – Coefficient sous-jacent à un ensemble d'actions

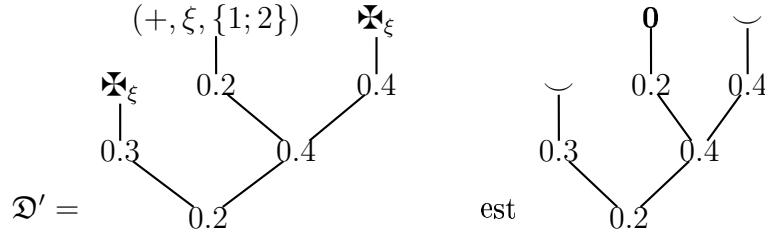
Le **coefficient sous-jacent** à un ensemble d'actions X dans un coefficient c est le coefficient c' obtenu en annulant les feuilles vides de c dont l'action n'est pas dans X .

Si \mathcal{D} est un dessein sur la base $\vdash \xi$, on note $\text{Coef}_{\mathfrak{X}, \mathfrak{X}_\xi}(\mathcal{D})$ le coefficient sous-jacent dans le premier coefficient (le coefficient à la racine) de \mathcal{D} de l'ensemble $\{\mathfrak{X}; \mathfrak{X}_\xi\}$. De même, on note $\text{Coef}_{pr}(\mathcal{D})$ le coefficient sous-jacent au premier coefficient de \mathcal{D} de l'ensemble des actions propres. Le premier coefficient, correspondant à l'ensemble des actions est noté $\text{Coef}_{fst}(\mathcal{D})$.

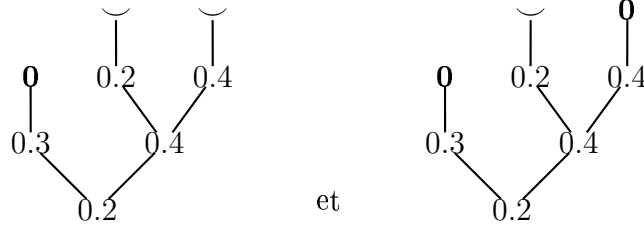
Par exemple, $\text{Coef}_{\mathfrak{X}, \mathfrak{X}_\xi}(\mathcal{D})$ où



Et $\text{Coef}_{\mathfrak{X}, \mathfrak{X}_\xi}(\mathcal{D}')$ où



Les coefficients sous-jacents $\text{Coef}_{pr}(\mathcal{D}')$ et $\text{Coef}_{pr}(\mathcal{D})$ sont respectivement



La définition de bi-éthique de la ludique simple (définition 1.64 page 45) est “probabilisée” en remplaçant les équivalences partielles par des pré-distances. L’idée générale est que deux desseins simples équivalents au sens des équivalences partielles sont à distance nulle d’eux-mêmes, deux desseins simples non équivalents sont à distance 1. Les combinaisons linéaires de desseins équivalents et non équivalents sont à des distances intermédiaires.

Définition 6.6 – Bi-éthique

Une **bi-éthique** est une paire (E, d) telle que :

- E est une éthique.
- d est une pré-distance sur les desseins partiels de E telle que :
 - Cas positif.
 - $d(\mathfrak{f}id, \mathfrak{f}id) = 0$
 - $d(\mathfrak{X}, \mathfrak{X}) = 0$ et $d(\mathfrak{X}_\xi, \mathfrak{X}_\xi) = 0$
 - $\forall \mathcal{D}_1, \mathcal{D}_2 \in EP, d(\mathcal{D}_1, \mathcal{D}_2) \geq \delta_c(\text{Coef}_{fst}(\mathcal{D}_1), \text{Coef}_{fst}(\mathcal{D}_2))$
 - $\forall \mathcal{D}_1, \mathcal{D}_2 \in EP, d(\mathcal{D}_1, \mathcal{D}_2) \geq \delta_c(\text{Coef}_{\mathfrak{X}, \mathfrak{X}_\xi}(\mathcal{D}_1), \text{Coef}_{\mathfrak{X}, \mathfrak{X}_\xi}(\mathcal{D}_2))$
 - Cas négatif.
 - $\forall \mathcal{D}_1, \mathcal{D}_2 \in EP, d(\mathcal{D}_1, \mathcal{D}_2) \geq \delta_c(\text{Coef}_{fst}(\mathcal{D}_1), \text{Coef}_{fst}(\mathcal{D}_2))$
 - $d(\mathcal{D}ai^-, \mathcal{D}ai^-) = 0$
 - $d(\mathfrak{G}kunk, \mathfrak{G}kunk) = 0$

Remarque 6.3 On remarque quelques similitudes avec les conditions de la définition de bi-éthique simple (définition 1.64 page 45) :

- Cas 1. La base β est positive.
 - Si $\mathcal{D} \equiv \mathcal{D}'$ alors :
 - si $\mathcal{D} = \mathfrak{f}id$ alors $\mathcal{D}' = \mathfrak{f}id$;
 - si $\mathcal{D} = \mathfrak{X}$ alors $\mathcal{D}' = \mathfrak{X}$ et si $\mathcal{D} = \mathfrak{X}_\xi$ alors $\mathcal{D}' = \mathfrak{X}_\xi$.
- Cas 2. La base β est négative.
 - $\mathfrak{G}kunk \equiv \mathfrak{G}kunk$ et $\mathcal{D}ai^- \equiv \mathcal{D}ai^-$.

Dans le cadre probabiliste, il y a quatre conditions pour le cas positif et trois conditions pour le cas négatif. La condition sur \mathfrak{Fid} est automatiquement vérifiée alors que les autres conditions sont duales l'une de l'autre (la deuxième condition (sur \mathfrak{X}_ξ) pour le cas positif est duale de la première condition pour le cas négatif...).

Les conditions $d(\mathfrak{S}\mathfrak{u}\mathfrak{n}\mathfrak{k}, \mathfrak{S}\mathfrak{u}\mathfrak{n}\mathfrak{k}) = 0$ et $d(\mathfrak{D}\mathfrak{a}\mathfrak{i}^-, \mathfrak{D}\mathfrak{a}\mathfrak{i}^-) = 0$ dans le cas négatif sont très similaires à celles utilisées en ludique simple : une condition d'équivalence en ludique simple ($\mathfrak{S}\mathfrak{u}\mathfrak{n}\mathfrak{k} \equiv \mathfrak{S}\mathfrak{u}\mathfrak{n}\mathfrak{k}$ et $\mathfrak{D}\mathfrak{a}\mathfrak{i}^- \equiv \mathfrak{D}\mathfrak{a}\mathfrak{i}^-$) est transformée en une condition de distance nulle ($d(\mathfrak{S}\mathfrak{u}\mathfrak{n}\mathfrak{k}, \mathfrak{S}\mathfrak{u}\mathfrak{n}\mathfrak{k}) = 0$ et $d(\mathfrak{D}\mathfrak{a}\mathfrak{i}^-, \mathfrak{D}\mathfrak{a}\mathfrak{i}^-) = 0$) en ludique probabiliste. Les autres conditions dans le cas négatif sont les conditions duales des conditions respectives du cas positif. Elles n'impliquent rien pour la ludique simple car le coefficient à la racine du traduit d'un dessein négatif différent de \mathfrak{Fid}^- est 1.

Les conditions sur la divergence $d(\mathfrak{F}\mathfrak{i}\mathfrak{d}, \mathfrak{F}\mathfrak{i}\mathfrak{d}) = 0$ ($\mathfrak{F}\mathfrak{i}\mathfrak{d} \equiv \mathfrak{F}\mathfrak{i}\mathfrak{d}$) et le démon $d(\mathfrak{X}, \mathfrak{X}) = 0$ ($\mathfrak{X} \equiv \mathfrak{X}$) ne sont pas explicitement données par Jean-Yves Girard mais sont des conséquences de l'orthogonalité. Les autres conditions dans le cas positif sont les conditions duales des conditions respectives du cas négatif. De plus, elles imposent que $\mathfrak{F}\mathfrak{i}\mathfrak{d} \equiv \mathfrak{D} \Rightarrow \mathfrak{D} = \mathfrak{F}\mathfrak{i}\mathfrak{d}$ et de même pour le démon.

Définition 6.7 – Orthogonal d'une bi-éthique

L'orthogonal d'une bi-éthique (E, d) sur une base unaire est la paire (E^\perp, d^\perp) où la pré-distance d^\perp est définie par

$$\forall \mathfrak{E}_1, \mathfrak{E}_2 \in E^{\perp P}, d^\perp(\mathfrak{E}_1, \mathfrak{E}_2) = \sup_{\mathfrak{D}_1, \mathfrak{D}_2 \in \text{Simp}(E^P)} (\delta_c(\llbracket \mathfrak{D}_1, \mathfrak{E}_1 \rrbracket, \llbracket \mathfrak{D}_2, \mathfrak{E}_2 \rrbracket) - d(\mathfrak{D}_1, \mathfrak{D}_2))$$

Sur une base quelconque $\xi_1^{\epsilon_1}, \dots, \xi_n^{\epsilon_n}$, une bi-éthique (E, d) est définie par orthogonalité (définition 6.17 page 160) :

$$\forall \mathfrak{D}, \mathfrak{D}' \in E^P, d(\mathfrak{D}, \mathfrak{D}') = \sup \delta_c(\llbracket \mathfrak{D}, (\mathfrak{E}_{\xi_i}) \rrbracket, \llbracket \mathfrak{D}', (\mathfrak{E}'_{\xi_i}) \rrbracket) - \sum_i d_i(\mathfrak{E}_{\xi_i}, \mathfrak{E}'_{\xi_i})$$

Dans la formule pour définir l'orthogonal d'une bi-éthique, le sup ne parcourt pas tous les desseins partiels mais simplement les desseins partiels simples. Cette restriction aux desseins simples est utile pour définir simplement des connecteurs comme les décalages par exemple (définition 6.12 page 157). Ainsi, il n'est pas nécessaire de définir une pré-distance sur tous les desseins mais uniquement sur les desseins simples. La pré-distance sur les autres desseins est calculée par orthogonalité.

Proposition 6.3 – Orthogonal d'une bi-éthique

L'orthogonal d'une bi-éthique est une bi-éthique.

Démonstration : Il suffit de montrer que

– d^\perp est bien une pré-distance.

Le point important est de vérifier que

$$\forall \mathfrak{D}_1, \mathfrak{D}_2, d^\perp(\mathfrak{D}_1, \mathfrak{D}_2) \geq 0$$

Pour cela, il suffit de trouver un dessin à distance nulle de lui-même dans chaque bi-éthique. On prend naturellement \mathfrak{Fid} pour les bi-éthiques positives et \mathfrak{Stunk} pour les bi-éthiques négatives.

$$\begin{aligned} d^\perp(\mathfrak{E}_1, \mathfrak{E}_2) &= \sup \delta_c(\llbracket \mathfrak{D}_1, \mathfrak{E}_1 \rrbracket, \llbracket \mathfrak{D}_2, \mathfrak{E}_2 \rrbracket) - d(\mathfrak{D}_1, \mathfrak{D}_2) \\ &\geq \delta_c(\llbracket \mathfrak{Fid}, \mathfrak{E}_1 \rrbracket, \llbracket \mathfrak{Fid}, \mathfrak{E}_2 \rrbracket) \geq 0 \end{aligned}$$

et

$$\begin{aligned} d^\perp(\mathfrak{E}_1, \mathfrak{E}_2) &= \sup \delta_c(\llbracket \mathfrak{D}_1, \mathfrak{E}_1 \rrbracket, \llbracket \mathfrak{D}_2, \mathfrak{E}_2 \rrbracket) - d(\mathfrak{D}_1, \mathfrak{D}_2) \\ &\geq \delta_c(\llbracket \mathfrak{Stunk}, \mathfrak{E}_1 \rrbracket, \llbracket \mathfrak{Stunk}, \mathfrak{E}_2 \rrbracket) \geq 0 \end{aligned}$$

- Cas positif.

- Montrons que $d^\perp(\mathfrak{Fid}, \mathfrak{Fid}) = 0$.

On a déjà remarqué $d^\perp(\mathfrak{Fid}, \mathfrak{Fid}) \geq 0$.

Il suffit donc de prouver que $d^\perp(\mathfrak{Fid}, \mathfrak{Fid}) \leq 0$.

$$\begin{aligned} d^\perp(\mathfrak{Fid}, \mathfrak{Fid}) &= \sup \delta_c(\llbracket \mathfrak{E}_1, \mathfrak{Fid} \rrbracket, \llbracket \mathfrak{E}_2, \mathfrak{Fid} \rrbracket) - d(\mathfrak{E}_1, \mathfrak{E}_2) \\ &= \sup \delta_c(\mathbf{0}, \mathbf{0}) - d(\mathfrak{E}_1, \mathfrak{E}_2) \\ &= \sup 0 - d(\mathfrak{E}_1, \mathfrak{E}_2) \\ &\leq 0 \end{aligned}$$

Donc $d^\perp(\mathfrak{Fid}, \mathfrak{Fid}) \leq 0$

- Montrons que $d^\perp(\mathfrak{X}_\xi, \mathfrak{X}_\xi) = 0$.

On a déjà remarqué $d^\perp(\mathfrak{X}_\xi, \mathfrak{X}_\xi) \geq 0$.

Il suffit donc à prouver que $d^\perp(\mathfrak{X}_\xi, \mathfrak{X}_\xi) \leq 0$.

$$\begin{aligned} d^\perp(\mathfrak{X}_\xi, \mathfrak{X}_\xi) &= \sup \delta_c(\llbracket \mathfrak{E}_1, \mathfrak{X}_\xi \rrbracket, \llbracket \mathfrak{E}_2, \mathfrak{X}_\xi \rrbracket) - d(\mathfrak{E}_1, \mathfrak{E}_2) \\ &= \sup \delta_c(\text{Coef}_{fst}(\mathfrak{E}_1), \text{Coef}_{fst}(\mathfrak{E}_2)) - d(\mathfrak{E}_1, \mathfrak{E}_2) \end{aligned}$$

Or pour tout \mathfrak{E}_1 et \mathfrak{E}_2 , $\delta_c(\text{Coef}_{fst}(\mathfrak{E}_1), \text{Coef}_{fst}(\mathfrak{E}_2)) - d(\mathfrak{E}_1, \mathfrak{E}_2) \leq 0$.

Donc $d^\perp(\mathfrak{X}_\xi, \mathfrak{X}_\xi) \leq 0$.

$$d^\perp(\mathfrak{X}, \mathfrak{X}) = \sup \delta_c(\llbracket \mathfrak{E}_1, \mathfrak{X} \rrbracket, \llbracket \mathfrak{E}_2, \mathfrak{X} \rrbracket) - d(\mathfrak{E}_1, \mathfrak{E}_2)$$

$$\begin{aligned} \text{De même,} \quad &= \sup \delta_c(\mathbf{1}, \mathbf{1}) - d(\mathfrak{E}_1, \mathfrak{E}_2) \\ &\leq 0 \end{aligned}$$

- Montrons $d^\perp(\mathfrak{D}_1, \mathfrak{D}_2) \geq \delta_c(\text{Coef}_{fst}(\mathfrak{D}_1), \text{Coef}_{fst}(\mathfrak{D}_2))$.

$$\begin{aligned} d^\perp(\mathfrak{D}_1, \mathfrak{D}_2) &\geq \delta_c(\llbracket \mathfrak{D}_1, \mathfrak{Dai}^- \rrbracket, \llbracket \mathfrak{D}_2, \mathfrak{Dai}^- \rrbracket) - d(\mathfrak{Dai}^-, \mathfrak{Dai}^-) \\ &= \delta_c(\text{Coef}_{fst}(\mathfrak{D}_1), \text{Coef}_{fst}(\mathfrak{D}_2)) - d(\mathfrak{Dai}^-, \mathfrak{Dai}^-) \\ &= \delta_c(\text{Coef}_{fst}(\mathfrak{D}_1), \text{Coef}_{fst}(\mathfrak{D}_2)) \end{aligned}$$

- Montrons $d^\perp(\mathfrak{D}_1, \mathfrak{D}_2) \geq \delta_c(\text{Coef}_{\mathfrak{X}_\xi}(\mathfrak{D}_1), \text{Coef}_{\mathfrak{X}_\xi}(\mathfrak{D}_2))$.

$$\begin{aligned} d^\perp(\mathfrak{D}_1, \mathfrak{D}_2) &\geq \delta_c(\llbracket \mathfrak{D}_1, \mathfrak{Stunk} \rrbracket, \llbracket \mathfrak{D}_2, \mathfrak{Stunk} \rrbracket) - d(\mathfrak{Stunk}, \mathfrak{Stunk}) \\ &= \delta_c(\text{Coef}_{\mathfrak{X}_\xi}(\mathfrak{D}_1), \text{Coef}_{\mathfrak{X}_\xi}(\mathfrak{D}_2)) - d(\mathfrak{Stunk}, \mathfrak{Stunk}) \\ &= \delta_c(\text{Coef}_{\mathfrak{X}_\xi}(\mathfrak{D}_1), \text{Coef}_{\mathfrak{X}_\xi}(\mathfrak{D}_2)) \end{aligned}$$

- Cas négatif.

- Montrons $d^\perp(\mathfrak{D}_1, \mathfrak{D}_2) \geq \delta_c(\text{Coef}_{fst}(\mathfrak{D}_1), \text{Coef}_{fst}(\mathfrak{D}_2))$.

$$\begin{aligned} d^\perp(\mathfrak{D}_1, \mathfrak{D}_2) &\geq \delta_c(\llbracket \mathfrak{D}_1, \mathfrak{X}_\xi \rrbracket, \llbracket \mathfrak{D}_2, \mathfrak{X}_\xi \rrbracket) - d(\mathfrak{X}_\xi, \mathfrak{X}_\xi) \\ &= \delta_c(\text{Coef}_{fst}(\mathfrak{D}_1), \text{Coef}_{fst}(\mathfrak{D}_2)) - d(\mathfrak{X}_\xi, \mathfrak{X}_\xi) \\ &= \delta_c(\text{Coef}_{fst}(\mathfrak{D}_1), \text{Coef}_{fst}(\mathfrak{D}_2)) \end{aligned}$$

- Montrons $d^\perp(\mathfrak{Dai}^-, \mathfrak{Dai}^-) = 0$

$$\begin{aligned} d^\perp(\mathfrak{Dai}^-, \mathfrak{Dai}^-) &= \sup \delta_c(\llbracket \mathfrak{D}_1, \mathfrak{Dai}^- \rrbracket - \llbracket \mathfrak{D}_2, \mathfrak{Dai}^- \rrbracket) - d(\mathfrak{D}_1, \mathfrak{D}_2) \\ &= \sup \delta_c(\text{Coef}_{fst}(\mathfrak{D}_1), \text{Coef}_{fst}(\mathfrak{D}_2)) - d(\mathfrak{D}_1, \mathfrak{D}_2) \\ &= 0 \end{aligned}$$

- Montrons $d^\perp(\mathfrak{Stunk}, \mathfrak{Stunk}) = 0$.

$$\begin{aligned} d^\perp(\mathfrak{Stunk}, \mathfrak{Stunk}) &= \sup \delta_c(\llbracket \mathfrak{D}_1, \mathfrak{Stunk} \rrbracket - \llbracket \mathfrak{D}_2, \mathfrak{Stunk} \rrbracket) - d(\mathfrak{D}_1, \mathfrak{D}_2) \\ &= \sup \delta_c(\text{Coef}_{\mathfrak{X}_\xi}(\mathfrak{D}_1), \text{Coef}_{\mathfrak{X}_\xi}(\mathfrak{D}_2)) - d(\mathfrak{D}_1, \mathfrak{D}_2) \\ &= 0 \end{aligned}$$

■

Définition 6.8 – Bi-comportement

Un **bi-comportement** est une bi-éthique égale à son bi-orthogonal.

La proposition suivante donne un moyen très simple de construire un bi-comportement.

Proposition 6.4 – Orthogonal d'une bi-éthique et bi-comportement

L'orthogonal d'une bi-éthique est un bi-comportement.

De plus, pour toute bi-éthique (E, d) et tous desseins simples \mathfrak{D}_1 et \mathfrak{D}_2 de E ,

$$d^{\perp\perp}(\mathfrak{D}_1, \mathfrak{D}_2) \leq d(\mathfrak{D}_1, \mathfrak{D}_2)$$

Démonstration: Soit (E, d) une bi-éthique. Montrons que $(E^\perp, d^\perp) = (E^{\perp\perp\perp}, d^{\perp\perp\perp})$. On a bien $E^\perp = E^{\perp\perp\perp}$. Il reste à prouver que $d^\perp = d^{\perp\perp\perp}$. Pour cela, on considère l'ordre \leq_d entre pré-distances comme l'extension point par point de l'ordre standard sur $[0; 1]$.

1. On a $\forall d_1 d_2. d_1 \leq_d d_2 \implies d_2^\perp \leq_d d_1^\perp$. Cela vient de la contra-variance à droite de l'opérateur $-$ dans les réels et de la co-variance du sup.
2. On a $\forall d. d^{\perp\perp} \leq_d d$. Car pour tous \mathfrak{E}_1 et \mathfrak{E}_2 simples,

$$\begin{aligned} & d^{\perp\perp}(\mathfrak{E}_1, \mathfrak{E}_2) \\ &= \sup_{\mathfrak{D}_1, \mathfrak{D}_2} \delta_c(\llbracket \mathfrak{D}_1, \mathfrak{E}_1 \rrbracket, \llbracket \mathfrak{D}_2, \mathfrak{E}_2 \rrbracket) - d^\perp(\mathfrak{D}_1, \mathfrak{D}_2) \\ &= \sup_{\mathfrak{D}_1, \mathfrak{D}_2} \delta_c(\llbracket \mathfrak{D}_1, \mathfrak{E}_1 \rrbracket, \llbracket \mathfrak{D}_2, \mathfrak{E}_2 \rrbracket) \\ &\quad - (\sup_{\mathfrak{E}'_1, \mathfrak{E}'_2} \delta_c(\llbracket \mathfrak{D}_1, \mathfrak{E}'_1 \rrbracket, \llbracket \mathfrak{D}_2, \mathfrak{E}'_2 \rrbracket) - d(\mathfrak{E}'_1, \mathfrak{E}'_2)) \\ &\leq \sup_{\mathfrak{D}_1, \mathfrak{D}_2} \delta_c(\llbracket \mathfrak{D}_1, \mathfrak{E}_1 \rrbracket, \llbracket \mathfrak{D}_2, \mathfrak{E}_2 \rrbracket) \\ &\quad - (\delta_c(\llbracket \mathfrak{D}_1, \mathfrak{E}_1 \rrbracket, \llbracket \mathfrak{D}_2, \mathfrak{E}_2 \rrbracket) - d(\mathfrak{E}_1, \mathfrak{E}_2)) \\ &= d(\mathfrak{E}_1, \mathfrak{E}_2) \end{aligned}$$

On a, par (2) $d^{\perp\perp} \leq_d d$ donc par (1), $d^\perp \leq_d d^{\perp\perp\perp}$. Par (2) (appliqué à d^\perp), on a aussi $d^{\perp\perp\perp} \leq_d d^\perp$. On conclut $d^\perp = d^{\perp\perp\perp}$. ■

Définition 6.9 – Dessein uniforme

Un dessein \mathfrak{D} est **uniforme** dans un bi-comportement (G, d) si

$$d(\mathfrak{D}, \mathfrak{D}) = 0$$

Remarque 6.4 En ludique simple, le bi-orthogonal $\equiv^{\perp\perp}$ d'une équivalence partielle \equiv contient \equiv . Donc le passage au bi-orthogonal "augmente" l'uniformité. Similairement, en ludique probabiliste, le passage au bi-orthogonal baisse la distance ce qui "augmente" l'uniformité.

Définition 6.10 – Bi-incarnation

La **bi-incarnation** d'un dessein uniforme \mathfrak{D} dans un bi-comportement (G, d) est

$$\|\mathfrak{D}\|_G = \bigcap \{ \mathfrak{D}' \mid \mathfrak{D}' \subseteq \mathfrak{D}, d(\mathfrak{D}, \mathfrak{D}') = 0 \}$$

Un dessein égal à sa bi-incarnation est dit **bi-matériel** ou **bi-incarné**.

Remarque 6.5 *L'hypothèse d'uniformité de \mathcal{D} est utile à l'existence de la bi-incarnation (sinon, l'ensemble $\{\mathcal{D}' \mid \mathcal{D}' \subseteq \mathcal{D}, d(\mathcal{D}, \mathcal{D}') = 0\}$ peut être vide.*

6.3 Connecteurs

De même qu'en ludique simple (section 1.7.2 page 46), il faut naturellement définir l'action des connecteurs (section 5.2 page 145) sur les bi-comportements.

On se contente de définir les distances sur les desseins simples. La définition sur les autres desseins est faite par bi-orthogonal. La notion utile est celle de bi-éthique complète.

Définition 6.11 – Bi-éthique complète

Une bi-éthique (E, d) est **complète** si

$$\text{Pss}(|E^{\perp\perp}|) \subseteq E$$

et pour tous les desseins simples

$$d(\mathcal{D}_1, \mathcal{D}_2) = d^{\perp\perp}(\mathcal{D}_1, \mathcal{D}_2)$$

6.3.1 Délocalisation et décalage

Définition 6.12 – Décalages

– Le **décalage positif** d'un bi-comportement négatif de base $\xi.i \vdash$ est

$$\downarrow(G, d) = (\downarrow G, \downarrow d) = (\downarrow G, \downarrow' d)^{\perp\perp}$$

où

$$\downarrow' d(\downarrow \mathcal{D}_1, \downarrow \mathcal{D}_2) = d(\mathcal{D}_1, \mathcal{D}_2)$$

– Le **décalage négatif** d'un bi-comportement positif de base $\vdash \xi.i$ est

$$\uparrow(G, d) = (\uparrow G, \uparrow d) = (\uparrow G, \uparrow' d)^{\perp\perp}$$

où

$$\uparrow' d(\dots \uparrow \mathcal{D}_1 \dots, \dots \uparrow \mathcal{D}_2 \dots) = d(\mathcal{D}_1, \mathcal{D}_2)$$

et $\dots \uparrow \mathcal{D}_k \dots$ est une notation pour un dessein négatif dont la continuation de $(-, \xi, \{i\})$ est \mathcal{D}_k .

Remarque 6.6 *La définition du décalage positif est valide car l'ensemble des décalages de desseins négatifs contient les desseins positifs simples propres du comportement $\downarrow G$. La définition du décalage négatif utilise les desseins sous la forme $\dots \uparrow \mathcal{D}_k \dots$ car le décalage négatif n'est pas surjectif sur les desseins négatifs.*

La proposition suivante simplifie la définition 6.12 page précédente car elle exprime que sur les desseins simples, le bi-orthogonal est inutile : on a une bi-éthique complète.

Proposition 6.5 – Décalages

Pour tout bi-comportement négatif (G, d) de base $\xi.i \vdash$, on a pour tous desseins simples

$$\downarrow d(\mathcal{D}_1, \mathcal{D}_2) = \downarrow d(\mathcal{D}_1, \mathcal{D}_2)$$

De même, pour tout bi-comportement positif (G, d) de base $\vdash \xi.i$, on a pour tous desseins simples

$$\uparrow d(\mathcal{D}_1, \mathcal{D}_2) = \uparrow d(\mathcal{D}_1, \mathcal{D}_2)$$

Démonstration : Il suffit de montrer que les pré-distances $\downarrow d$ et $\uparrow d$ sont des orthogonales (proposition 6.4 page 156). La preuve montre de plus qu'elles sont en fait orthogonales l'une de l'autre.

- Soit (G, d) un bi-comportement négatif de base $\xi.i \vdash$. Soient \mathcal{D}_1 et \mathcal{D}_2 deux desseins simples de G .

On a :

$$\begin{aligned} \downarrow d(\downarrow \mathcal{D}_1, \downarrow \mathcal{D}_2) &= d(\mathcal{D}_1, \mathcal{D}_2) \\ &= \sup_{\mathfrak{E}_1, \mathfrak{E}_2} \delta_c(\llbracket \mathcal{D}_1, \mathfrak{E}_1 \rrbracket, \llbracket \mathcal{D}_2, \mathfrak{E}_2 \rrbracket) - d^\perp(\mathfrak{E}_1, \mathfrak{E}_2) \\ &= \sup_{\mathfrak{E}_1, \mathfrak{E}_2} \delta_c(\llbracket \downarrow \mathcal{D}_1, \uparrow \mathfrak{E}_1 \rrbracket, \llbracket \downarrow \mathcal{D}_2, \uparrow \mathfrak{E}_2 \rrbracket) - (\uparrow d)^\perp(\uparrow \mathfrak{E}_1, \uparrow \mathfrak{E}_2) \\ &= \sup_{\mathfrak{E}'_1, \mathfrak{E}'_2} \delta_c(\llbracket \downarrow \mathcal{D}_1, \mathfrak{E}'_1 \rrbracket, \llbracket \downarrow \mathcal{D}_2, \mathfrak{E}'_2 \rrbracket) - (\uparrow d)^\perp(\mathfrak{E}'_1, \mathfrak{E}'_2) \end{aligned}$$

Donc

$$\downarrow d(\downarrow \mathcal{D}_1, \downarrow \mathcal{D}_2) = \downarrow d(\downarrow \mathcal{D}_1, \downarrow \mathcal{D}_2)$$

On conclut par la surjectivité du décalage positif sur les desseins simples.

- On procède de même dans le décalage négatif.

On obtient

$$\uparrow d(\dots \uparrow \mathcal{D}_1 \dots, \dots \uparrow \mathcal{D}_2 \dots) = \uparrow d(\dots \uparrow \mathcal{D}_1 \dots, \dots \uparrow \mathcal{D}_2 \dots) \quad \blacksquare$$

6.3.2 Additifs

Définition 6.13 – Intersection

L'intersection de bi-comportements négatifs de même base est

$$\cap_k(G_k, d_k) = (\cap_k G_k, \cap_k d_k)$$

avec

$$\cap_k d_k = \sup d_k$$

Proposition 6.6 – Intersection de bi-comportements

L'intersection de bi-comportements de base β est un bi-comportement de base β .

Définition 6.14 – Borne inférieure sur des domaines distincts

Soit B un treillis complet. Soit (f_i) une famille de fonctions $f_i : A_i \rightarrow B$. Soit $A = \cup A_i$. La **borne inférieure** de la famille (f_i) est

$$\inf f_i : A \rightarrow B \\ x \mapsto \inf_{x \in A_i} f_i(x)$$

Définition 6.15 – Union

L'union de bi-comportements positifs de même base est

$$\cup_k^* (G_k, d_k) = (\cup_k^* G_k, \cup_k^* d_k) = (\cup_k^* G_k, \cup_k d_k)^{\perp\perp}$$

avec

$$\cup_k d_k = \inf d_k$$

Ces définitions (*intersection* et *union*) définissent les connecteurs *avec* et *plus* sur les comportements disjoints.

6.3.3 Multiplicatifs

La notion de multiplicatif négatif (le *par* en logique linéaire) est reliée à celle d'implication et de négation à cause de la formule de la logique linéaire

$$A \multimap B = A^\perp \wp B$$

Ainsi la définition de *nœud papillon* est proche de la définition 6.7 page 154 d'orthogonal d'une bi-éthique.

Définition 6.16 – Le nœud papillon

Le connecteur \bowtie de deux bi-comportements négatifs de même base (G_1, d_1) et (G_2, d_2) est :

$$(G_1, d_1) \bowtie (G_2, d_2) = (G_1 \bowtie G_2, d_1 \bowtie d_2)$$

avec

$$(d_1 \bowtie d_2)(\mathfrak{F}_1, \mathfrak{F}_2) = \sup d_2((\mathfrak{F}_1)\mathfrak{D}_1, (\mathfrak{F}_2)\mathfrak{D}_2) - d_1^\perp(\mathfrak{D}_1, \mathfrak{D}_2)$$

Proposition 6.7 – Le nœud papillon

Pour tous bi-comportements négatifs de même base (G_1, d_1) et (G_2, d_2) , la bi-éthique $(G_1 \bowtie G_2, d_1 \bowtie d_2)$ est un bi-comportement.

Démonstration : Il suffit (proposition 6.4 page 156) de montrer que c'est l'orthogonal d'une bi-éthique. C'est l'orthogonal des bi-éthiques $(G_1^\perp \otimes G_2^\perp, d')$ telles que sur les desseins simples :

$$d'(\mathfrak{D}_1^0, \mathfrak{D}_2^0) = \min\{1; \inf_{\forall i \mathfrak{D}_i^0 = \mathfrak{D}_i \otimes \mathfrak{D}_i'} d_1(\mathfrak{D}_1, \mathfrak{D}_2) + d_2(\mathfrak{D}_1', \mathfrak{D}_2')\}$$

En effet,

$$\begin{aligned}
 (d_1 \bowtie d_2)(\mathfrak{F}_1, \mathfrak{F}_2) &= \sup d_2((\mathfrak{F}_1)\mathcal{D}_1, (\mathfrak{F}_2)\mathcal{D}_2) - d_1^\perp(\mathcal{D}_1, \mathcal{D}_2) \\
 &= \sup \delta_c(\llbracket (\mathfrak{F}_1)\mathcal{D}_1, \mathcal{D}'_1 \rrbracket, \llbracket (\mathfrak{F}_2)\mathcal{D}_2, \mathcal{D}'_2 \rrbracket) - d_2^\perp(\mathcal{D}'_1, \mathcal{D}'_2) - d_1^\perp(\mathcal{D}_1, \mathcal{D}_2) \\
 &= \sup \delta_c(\llbracket \mathfrak{F}_1, \mathcal{D}_1 \otimes \mathcal{D}'_1 \rrbracket, \llbracket \mathfrak{F}_2, \mathcal{D}_2 \otimes \mathcal{D}'_2 \rrbracket) - (d_1^\perp(\mathcal{D}_1, \mathcal{D}_2) + d_2^\perp(\mathcal{D}'_1, \mathcal{D}'_2)) \\
 &= \sup \delta_c(\llbracket \mathfrak{F}_1, \mathcal{D}_1 \otimes \mathcal{D}'_1 \rrbracket, \llbracket \mathfrak{F}_2, \mathcal{D}_2 \otimes \mathcal{D}'_2 \rrbracket) - d'(\mathcal{D}_1 \otimes \mathcal{D}'_1, \mathcal{D}_2 \otimes \mathcal{D}'_2)
 \end{aligned}$$

Il ne reste qu'à invoquer la surjectivité du tenseur sur les desseins simples. ■

Remarque 6.7 *On a utilisé le fait que la connaissance de la distance sur les desseins simples suffit à définir l'orthogonal.*

Proposition 6.8 – Le *nœud papillon* est commutatif

Soient (G_1, d_1) et (G_2, d_2) deux bi-comportements négatifs de même base. On a :

$$\begin{aligned}
 (d_1 \bowtie d_2)(\mathfrak{F}_1, \mathfrak{F}_2) &= \sup d_2((\mathfrak{F}_1)\mathcal{D}_1, (\mathfrak{F}_2)\mathcal{D}_2) - d_1^\perp(\mathcal{D}_1, \mathcal{D}_2) \\
 &= \sup d_1((\mathfrak{F}_1)\mathcal{D}_1, (\mathfrak{F}_2)\mathcal{D}_2) - d_2^\perp(\mathcal{D}_1, \mathcal{D}_2)
 \end{aligned}$$

Démonstration : D'après la preuve de la proposition 6.8 , on a :

$$\begin{aligned}
 (d_1 \bowtie d_2)(\mathfrak{F}_1, \mathfrak{F}_2) &= \sup \delta_c(\llbracket \mathfrak{F}_1, \mathcal{D}_1, \mathcal{D}'_1 \rrbracket, \llbracket \mathfrak{F}_2, \mathcal{D}_2, \mathcal{D}'_2 \rrbracket) - (d_1^\perp(\mathcal{D}_1, \mathcal{D}_2) + d_2^\perp(\mathcal{D}'_1, \mathcal{D}'_2))
 \end{aligned}$$

Cette expression est visiblement symétrique en d_1 et d_2 . ■

Définition 6.17 – Séquent de bi-comportements

Un **séquent de bi-comportements** sur une base $\Xi \vdash \Lambda$ notée $(\xi_i^{\epsilon_i})$ est un ensemble Ξ, Λ de bi-comportements (G_i, d_i) sur les bases $\vdash \xi_i$.

Le bi-comportement $\Xi \vdash \Lambda$ sur la base $\Xi \vdash \Lambda$ est l'orthogonal des familles de desseins (\mathfrak{E}_{ξ_i}) où $\mathfrak{E}_{\xi_i} \in G_i$ si $\xi_i \in \Xi$ et $\mathfrak{E}_{\xi_i} \in G_i^\perp$ sinon, avec la pré-distance d :

$$d(\mathfrak{D}, \mathfrak{D}') = \sup \delta_c(\llbracket \mathfrak{D}, (\mathfrak{E}_{\xi_i}) \rrbracket, \llbracket \mathfrak{D}', (\mathfrak{E}'_{\xi_i}) \rrbracket) - \sum_i d_i^{\epsilon_i \perp}(\mathfrak{E}_{\xi_i}, \mathfrak{E}'_{\xi_i})$$

$$\text{où } d^{\epsilon \perp} = \begin{cases} d & \text{si } \epsilon = - \\ d^\perp & \text{si } \epsilon = + \end{cases}$$

6.4 Complétude vis-à-vis de la ludique simple

La notion suivante est la clef pour la complétude car elle permet de relier l'uniformité en ludique probabiliste à l'uniformité en ludique simple.

Définition 6.18 – Bi-comportement s-régulier

Un bi-comportement probabiliste (G, d_G) est **s-régulier** pour un bi-comportement simple (H, \equiv) si

- G est s-régulier pour H .
- pour tous desseins simples \mathcal{D}_1 et \mathcal{D}_2 de H^p

$$d_G(\mathcal{D}_1, \mathcal{D}_2) = 0 \Leftrightarrow \mathcal{D}_1 \equiv \mathcal{D}_2$$

$$d_G(\mathcal{D}_1, \mathcal{D}_2) = 1 \Leftrightarrow \mathcal{D}_1 \not\equiv \mathcal{D}_2$$

- pour tous desseins simples \mathcal{E}_1 et \mathcal{E}_2 de $H^{\perp_s p}$

$$d_G(\mathcal{E}_1, \mathcal{E}_2) = 0 \Leftrightarrow \mathcal{E}_1 \equiv^\perp \mathcal{E}_2$$

$$d_G(\mathcal{E}_1, \mathcal{E}_2) = 1 \Leftrightarrow \mathcal{E}_1 \not\equiv^\perp \mathcal{E}_2$$

Un bi-comportement probabiliste est **s-régulier** s'il l'est pour un bi-comportement simple donné.

Définition 6.19 – Connecteur s-régulier

Un connecteur est **s-régulier** s'il envoie des bi-comportements s-réguliers sur des bi-comportements s-réguliers.

Proposition 6.9 – Dual d'un bi-comportement s-régulier

L'orthogonal d'un bi-comportement s-régulier est s-régulier.

Démonstration : Soit (G, d) un bi-comportement s-régulier pour le bi-comportement simple (H, \equiv) .

Soient \mathcal{E} et \mathcal{E}' deux desseins simples de G^\perp .

- Cas 1. $\mathcal{E} \equiv^\perp \mathcal{E}'$.

On a

$$\mathcal{D} \equiv \mathcal{D}' \Rightarrow \llbracket \mathcal{E}, \mathcal{D} \rrbracket = \llbracket \mathcal{E}', \mathcal{D}' \rrbracket$$

donc

$$\mathcal{D} \equiv \mathcal{D}' \Rightarrow \delta_c(\llbracket \mathcal{E}, \mathcal{D} \rrbracket, \llbracket \mathcal{E}', \mathcal{D}' \rrbracket) = 0$$

donc

$$d(\mathcal{D}, \mathcal{D}') = 0 \Rightarrow \delta_c(\llbracket \mathcal{E}, \mathcal{D} \rrbracket, \llbracket \mathcal{E}', \mathcal{D}' \rrbracket) = 0$$

donc

$$\begin{aligned} d^\perp(\mathcal{E}, \mathcal{E}') &= \sup \delta_c(\llbracket \mathcal{E}, \mathcal{D} \rrbracket, \llbracket \mathcal{E}', \mathcal{D}' \rrbracket) - d(\mathcal{D}, \mathcal{D}') \\ &= 0 \end{aligned}$$

- Cas 2. $\mathcal{E} \not\equiv^\perp \mathcal{E}'$. Alors, il existe $\mathcal{D} \equiv \mathcal{D}'$ simples tels que

$$\llbracket \mathcal{E}, \mathcal{D} \rrbracket \neq \llbracket \mathcal{E}', \mathcal{D}' \rrbracket$$

Dans ce cas, on a

$$\delta_c(\llbracket \mathcal{E}, \mathcal{D} \rrbracket, \llbracket \mathcal{E}', \mathcal{D}' \rrbracket) - d(\mathcal{D}, \mathcal{D}') = 1 - 0 = 1$$

On en déduit

$$\begin{aligned} d(\mathfrak{E}, \mathfrak{E}') &= \sup \delta_c(\llbracket \mathfrak{E}, \mathfrak{D} \rrbracket, \llbracket \mathfrak{E}', \mathfrak{D}' \rrbracket) - d(\mathfrak{D}, \mathfrak{D}') \\ &= 1 \end{aligned}$$

■

On en déduit immédiatement (par les lois de De Morgan) la proposition suivante.

Proposition 6.10 – Dual d'un connecteur s-régulier

Le dual d'un connecteur s-régulier est s-régulier.

Définition 6.20 – Bi-orthogonal probabiliste d'un bi-comportement simple

Le bi-orthogonal probabiliste d'un bi-comportement simple (G, \equiv) est

$$(G, \equiv)^{\perp_p \perp_p} = (G^{\perp_p \perp_p}, d_{\equiv}^{\perp \perp})$$

avec, pour tous desseins simples \mathfrak{D}_1 et \mathfrak{D}_2 ,

$$\begin{aligned} d_{\equiv}(\mathfrak{D}_1, \mathfrak{D}_2) &= 0 \quad \text{si } \mathfrak{D}_1 \equiv \mathfrak{D}_2 \\ &= 1 \quad \text{sinon} \end{aligned}$$

Remarque 6.8 La fonction d_{\equiv} est bien une pré-distance donc la définition précédente a bien un sens.

Proposition 6.11 – Bi-orthogonal probabiliste d'un bi-comportement simple

Pour tout comportement simple (G, \equiv) , son bi-orthogonal probabiliste $(G^{\perp_p \perp_p}, d_{\equiv}^{\perp \perp})$ est s-régulier.

Démonstration : Il suffit de montrer que sur les desseins simples,

$$d(\mathfrak{D}_1, \mathfrak{D}_2) = d_{\equiv}^{\perp \perp}(\mathfrak{D}_1, \mathfrak{D}_2)$$

C'est vrai car, sur les desseins simples, d_{\equiv} est orthogonal à la pré-distance d_{\equiv}^{\perp} où d_{\equiv}^{\perp} est la pré-distance correspondant à $(G, \equiv)^{\perp}$. On conclut en invoquant la proposition 6.4 page 156. ■

6.4.1 Décalages

Proposition 6.12 – La s-régularité des décalages

Les décalages sont s-réguliers.

Démonstration : Soit (G, d_G) un bi-comportement probabiliste de base $\vdash \xi$, s-régulier pour un bi-comportement simple (H, \equiv) .

– *Cas 1.* G et H sont de base $\xi.i \vdash$.

On a

$$\downarrow(G, d) = (\downarrow G, \downarrow d)$$

où

$$\downarrow d(\downarrow \mathcal{D}_1, \downarrow \mathcal{D}_2) = d(\mathcal{D}_1, \mathcal{D}_2)$$

et

$$\downarrow(H, \equiv) = (\downarrow H, \equiv_{\downarrow})$$

où

$$\downarrow \mathcal{D}_1 \equiv_{\downarrow} \downarrow \mathcal{D}_2 \iff \mathcal{D}_1 \equiv \mathcal{D}_2$$

On en déduit :

$$\downarrow \mathcal{D}_1 \equiv_{\downarrow} \downarrow \mathcal{D}_2 \iff \mathcal{D}_1 \equiv \mathcal{D}_2 \iff d(\mathcal{D}_1, \mathcal{D}_2) = 0 \iff \downarrow d(\downarrow \mathcal{D}_1, \downarrow \mathcal{D}_2) = 0$$

et

$$\downarrow \mathcal{D}_1 \not\equiv_{\downarrow} \downarrow \mathcal{D}_2 \iff \mathcal{D}_1 \not\equiv \mathcal{D}_2 \iff d(\mathcal{D}_1, \mathcal{D}_2) = 1 \iff \downarrow d(\downarrow \mathcal{D}_1, \downarrow \mathcal{D}_2) = 1$$

– *Cas 2.* G et H sont de base $\vdash \xi.i$.

On utilise la proposition 6.10 page ci-contre. ■

6.4.2 Additifs

Proposition 6.13 – La s -régularité du avec

Le connecteur $\&$ est s -régulier.

Démonstration : Soient (G_1, d_{G_1}) et (G_2, d_{G_2}) deux bi-comportements s -réguliers pour (H_1, \equiv_{H_1}) et (H_2, \equiv_{H_2}) .

Soient \mathcal{D}_1 et \mathcal{D}_2 deux desseins simples.

– Supposons $\mathcal{D}_1 \equiv_{H_1 \& H_2} \mathcal{D}_2$. On a alors $\mathcal{D}_1 \equiv_{H_1} \mathcal{D}_2$ et $\mathcal{D}_1 \equiv_{H_2} \mathcal{D}_2$ donc, par s -régularité, $d_{G_1}(\mathcal{D}_1, \mathcal{D}_2) = 0$ et $d_{G_2}(\mathcal{D}_1, \mathcal{D}_2) = 0$ donc

$$d_{G_1 \& G_2}(\mathcal{D}_1, \mathcal{D}_2) = \sup\{0; 0\} = 0$$

– Réciproquement, supposons $d_{G_1 \& G_2}(\mathcal{D}_1, \mathcal{D}_2) = 0$.

On a $d_{G_1}(\mathcal{D}_1, \mathcal{D}_2) = 0$ et $d_{G_2}(\mathcal{D}_1, \mathcal{D}_2) = 0$ donc $\mathcal{D}_1 \equiv_{H_1} \mathcal{D}_2$ et $\mathcal{D}_1 \equiv_{H_2} \mathcal{D}_2$ donc $\mathcal{D}_1 \equiv_{H_1 \& H_2} \mathcal{D}_2$.

– Supposons $\mathcal{D}_1 \not\equiv_{H_1 \& H_2} \mathcal{D}_2$. On a alors $\mathcal{D}_1 \not\equiv_{H_1} \mathcal{D}_2$ ou $\mathcal{D}_1 \not\equiv_{H_2} \mathcal{D}_2$ donc, par s -régularité, $d_{G_1}(\mathcal{D}_1, \mathcal{D}_2) = 1$ ou $d_{G_2}(\mathcal{D}_1, \mathcal{D}_2) = 1$ donc

$$d_{G_1 \& G_2}(\mathcal{D}_1, \mathcal{D}_2) = \sup\{1; 1\} = 1$$

– Réciproquement, supposons $d_{G_1 \& G_2}(\mathcal{D}_1, \mathcal{D}_2) = 1$.

On a $d_{G_1}(\mathcal{D}_1, \mathcal{D}_2) = 1$ ou $d_{G_2}(\mathcal{D}_1, \mathcal{D}_2) = 1$ donc $\mathcal{D}_1 \not\equiv_{H_1} \mathcal{D}_2$ ou $\mathcal{D}_1 \not\equiv_{H_2} \mathcal{D}_2$ donc $\mathcal{D}_1 \not\equiv_{H_1 \& H_2} \mathcal{D}_2$. ■

Proposition 6.14 – La s -régularité du plus

Le connecteur \oplus est s -régulier.

Démonstration : On utilise la proposition 6.10 page précédente. ■

6.4.3 Multiplicatifs

Proposition 6.15 – Le tenseur

Soient (G, d_G) et (H, d_H) deux bi-comportements positifs. On a pour tous desseins simples :

$$d_{G \otimes H}(\mathcal{D}_1 \otimes \mathcal{D}_2, \mathcal{D}'_1 \otimes \mathcal{D}'_2) \leq \min\{1; d_G(\mathcal{D}_1, \mathcal{D}'_1) + d_H(\mathcal{D}_2, \mathcal{D}'_2)\}$$

Démonstration : Puisque les pré-distances sont bornées par 1, il suffit de vérifier que :

$$d_{G \otimes H}(\mathcal{D}_1 \otimes \mathcal{D}_2, \mathcal{D}'_1 \otimes \mathcal{D}'_2) \leq d_G(\mathcal{D}_1, \mathcal{D}'_1) + d_H(\mathcal{D}_2, \mathcal{D}'_2)$$

On a

$$\begin{aligned} d_{G \otimes H}(\mathcal{E}, \mathcal{E}') &= \sup d_{H^\perp}(\llbracket \mathcal{E} \rrbracket_{\mathcal{D}_1}, \llbracket \mathcal{E}' \rrbracket_{\mathcal{D}'_1}) - d_G(\mathcal{D}_1, \mathcal{D}'_1) \\ &= \sup \delta_c(\llbracket \mathcal{E} \rrbracket_{\mathcal{D}_1}, \llbracket \mathcal{E}' \rrbracket_{\mathcal{D}'_1}) - d_G(\mathcal{D}_1, \mathcal{D}'_1) \end{aligned}$$

donc pour tous $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}'_1$ et \mathcal{D}'_2 ,

$$\delta_c(\llbracket \mathcal{E} \rrbracket_{\mathcal{D}_1}, \llbracket \mathcal{E}' \rrbracket_{\mathcal{D}'_1}) - d_{G \otimes H}(\mathcal{E}, \mathcal{E}') \leq d_G(\mathcal{D}_1, \mathcal{D}'_1) + d_H(\mathcal{D}_2, \mathcal{D}'_2)$$

donc

$$\begin{aligned} d_{G \otimes H}(\mathcal{D}_1 \otimes \mathcal{D}_2, \mathcal{D}'_1 \otimes \mathcal{D}'_2) &= \sup \delta_c(\llbracket \mathcal{D}_1 \otimes \mathcal{D}_2 \rrbracket_{\mathcal{E}}, \llbracket \mathcal{D}'_1 \otimes \mathcal{D}'_2 \rrbracket_{\mathcal{E}'}) - d_{G \otimes H}(\mathcal{E}, \mathcal{E}') \\ &= \sup \delta_c(\llbracket \mathcal{D}_1 \rrbracket_{\mathcal{E}}, \llbracket \mathcal{D}'_1 \rrbracket_{\mathcal{E}'}) - d_{G \otimes H}(\mathcal{E}, \mathcal{E}') \\ &\leq d_G(\mathcal{D}_1, \mathcal{D}'_1) + d_H(\mathcal{D}_2, \mathcal{D}'_2) \end{aligned} \quad \blacksquare$$

Proposition 6.16 – La s-régularité du par

Le connecteur \mathfrak{P} est s-régulier.

Démonstration : Soit G et H deux comportements négatifs. On a :

$$\begin{aligned} (d_{G \otimes H}(\mathfrak{F}_1, \mathfrak{F}_2) = \sup d_{H^\perp}(\llbracket \mathfrak{F}_1 \rrbracket_{\mathcal{D}_1}, \llbracket \mathfrak{F}_2 \rrbracket_{\mathcal{D}_2}) - d_G(\mathcal{D}_1, \mathcal{D}_2) = 0) \\ \Leftrightarrow (d_{H^\perp}(\llbracket \mathfrak{F}_1 \rrbracket_{\mathcal{D}_1}, \llbracket \mathfrak{F}_2 \rrbracket_{\mathcal{D}_2}) = 1 \Rightarrow d_G(\mathcal{D}_1, \mathcal{D}_2) = 1) \\ \Leftrightarrow (\llbracket \mathfrak{F}_1 \rrbracket_{\mathcal{D}_1} \equiv_{H^\perp} \llbracket \mathfrak{F}_2 \rrbracket_{\mathcal{D}_2} \Rightarrow \llbracket \mathfrak{F}_1 \rrbracket_{\mathcal{D}_1} \equiv_{G^\perp} \llbracket \mathfrak{F}_2 \rrbracket_{\mathcal{D}_2}) \\ \Leftrightarrow (\llbracket \mathfrak{F}_1 \rrbracket_{\mathcal{D}_1} \equiv_G \llbracket \mathfrak{F}_2 \rrbracket_{\mathcal{D}_2} \Rightarrow \llbracket \mathfrak{F}_1 \rrbracket_{\mathcal{D}_1} \equiv_H \llbracket \mathfrak{F}_2 \rrbracket_{\mathcal{D}_2}) \\ \Leftrightarrow (\mathfrak{F}_1 \equiv_G \mathfrak{F}_2 \Rightarrow \mathfrak{F}_1 \equiv_H \mathfrak{F}_2) \end{aligned} \quad \blacksquare$$

Proposition 6.17 – La s-régularité du tenseur

Le connecteur \otimes est s-régulier.

Démonstration : On utilise la proposition 6.10 page 162. \blacksquare

6.4.4 Les quantificateurs

La s-régularité des quantificateurs est identique à celle des additifs.

6.4.5 Séquents de bi-comportements

Proposition 6.18 – La s-régularité des séquents de bi-comportements

Tout dessein simple \mathcal{D} d'un séquent de bi-comportements simples $\Xi \vdash \Lambda$ appartient au séquent de bi-comportements probabiliste $\Xi^{\perp p \perp p} \vdash \Lambda^{\perp p \perp p}$.

Pour tous desseins \mathcal{D}_1 et \mathcal{D}_2 dans $\Xi \vdash \Lambda$, on a

$$\begin{aligned} d_{\Xi^{\perp p \perp p} \vdash \Lambda^{\perp p \perp p}}(\mathcal{D}_1, \mathcal{D}_2) &= 0 \Leftrightarrow \mathcal{D}_1 \equiv \mathcal{D}_2 \\ d_{\Xi^{\perp p \perp p} \vdash \Lambda^{\perp p \perp p}}(\mathcal{D}_1, \mathcal{D}_2) &= 1 \Leftrightarrow \mathcal{D}_1 \not\equiv \mathcal{D}_2 \end{aligned}$$

Démonstration : En utilisant les notations de la définition 6.17 page 160, on a :

$$d_{\Xi^{\perp p \perp p} \vdash \Lambda^{\perp p \perp p}}(\mathcal{D}, \mathcal{D}') = \sup \delta_c(\llbracket \mathcal{D}, (\mathfrak{E}_{\xi_i}) \rrbracket, \llbracket \mathcal{D}', (\mathfrak{E}'_{\xi_i}) \rrbracket) - \sum_i d_{\Xi^{\perp p \perp p}}^{\epsilon_i \perp}(\mathfrak{E}_{\xi_i}, \mathfrak{E}'_{\xi_i})$$

– Soit $\mathcal{D} \equiv \mathcal{D}'$.

Par s-régularité des $\Xi^{\perp \perp}, \Lambda^{\perp \perp}$,

$$\forall i \ d_{\Xi^{\perp p \perp p}}^{\epsilon_i \perp}(\mathfrak{E}_{\xi_i}, \mathfrak{E}'_{\xi_i}) = 1 \text{ si } \mathfrak{E}_{\xi_i} \not\equiv_i^{\epsilon_i \perp} \mathfrak{E}'_{\xi_i}$$

Donc

$$\begin{aligned} d_{\Xi^{\perp p \perp p} \vdash \Lambda^{\perp p \perp p}}(\mathcal{D}, \mathcal{D}') &= \sup \delta_c(\llbracket \mathcal{D}, (\mathfrak{E}_{\xi_i}) \rrbracket, \llbracket \mathcal{D}', (\mathfrak{E}'_{\xi_i}) \rrbracket) - \sum_i d_{\Xi^{\perp p \perp p}}^{\epsilon_i \perp}(\mathfrak{E}_{\xi_i}, \mathfrak{E}'_{\xi_i}) \\ &= \sup_{d_{\Xi^{\perp p \perp p}}^{\epsilon_i \perp}(\mathfrak{E}_{\xi_i}, \mathfrak{E}'_{\xi_i})=0} \delta_c(\llbracket \mathcal{D}, (\mathfrak{E}_{\xi_i}) \rrbracket, \llbracket \mathcal{D}', (\mathfrak{E}'_{\xi_i}) \rrbracket) - 0 \\ &= \sup_{d_{\Xi^{\perp p \perp p}}^{\epsilon_i \perp}(\mathfrak{E}_{\xi_i}, \mathfrak{E}'_{\xi_i})=0} 0 - 0 \\ &= 0 \end{aligned}$$

– Soit $\mathcal{D} \not\equiv \mathcal{D}'$.

Soit (\mathfrak{E}_{ξ_i}) et (\mathfrak{E}'_{ξ_i}) tels que $\forall i, \mathfrak{E}_{\xi_i} \equiv_i^{\epsilon_i \perp} \mathfrak{E}'_{\xi_i}$ et $\llbracket \mathcal{D}, (\mathfrak{E}_{\xi_i}) \rrbracket \not\equiv \llbracket \mathcal{D}', (\mathfrak{E}'_{\xi_i}) \rrbracket$. On a :

$$\begin{aligned} d_{\Xi^{\perp p \perp p} \vdash \Lambda^{\perp p \perp p}}(\mathcal{D}, \mathcal{D}') &= \sup \delta_c(\llbracket \mathcal{D}, (\mathfrak{E}_{\xi_i}) \rrbracket, \llbracket \mathcal{D}', (\mathfrak{E}'_{\xi_i}) \rrbracket) - \sum_i d_{\Xi^{\perp p \perp p}}^{\epsilon_i \perp}(\mathfrak{E}_{\xi_i}, \mathfrak{E}'_{\xi_i}) \\ &\geq \delta_c(\llbracket \mathcal{D}, (\mathfrak{E}_{\xi_i}) \rrbracket, \llbracket \mathcal{D}', (\mathfrak{E}'_{\xi_i}) \rrbracket) - 0 \\ &= 1 - 0 \\ &= 1 \end{aligned} \quad \blacksquare$$

6.5 Complétude forte

La complétude forte est d'abord montrée vis-à-vis de la ludique simple. La complétude forte vis-à-vis de MALL_2 et MAAL_2 est obtenue par composition.

6.5.1 Condition de gain

On reprend directement les définitions de la ludique simple (section 1.9 page 47) et la notion de présence (définition 2.15 page 59).

Définition 6.21 – Obstination

| Un dessein est **obstiné** s'il n'utilise pas le démon.

Définition 6.22 – Dessein présent

| Un dessein est **présent** si le poids de chacun de ses coefficients négatifs est 1.

Définition 6.23 – c -tranche exacte

| Une c -tranche est **exacte** si elle est de la forme

- \boxtimes
- \boxtimes_ξ
- $(+, \xi, \emptyset)$
- $(+, \xi, \{i_1, \dots, i_n\}).(\mathcal{D}_{\xi.i_1}, \dots, \mathcal{D}_{\xi.i_n})$ où les $\mathcal{D}_{\xi.i_k}$ sont exacts.
- $(-, \xi, \{i_1, \dots, i_n\}).\mathcal{D}$ où \mathcal{D} est exact et les $\xi.i_k$ apparaissent dans \mathcal{D} .
- $(-, \xi, \{i_1, \dots, i_n\}).\mathcal{D}$ où \mathcal{D} est exact et infini.
- $c.(\mathcal{D}_1, \dots, \mathcal{D}_n)$ où les \mathcal{D}_i sont exacts.

Remarque 6.9 Une définition équivalente est qu'une c -tranche est exacte si et seulement si toutes ses projections simples sont exactes.

Définition 6.24 – Dessein exact

| Un dessein est **exact** si toutes ses c -tranches maximales pour l'inclusion sont exactes.

Définition 6.25 – Parcimonie

| Un dessein est **parcimonieux** si la tranche visitée durant toute normalisation avec un dessein orthogonal est exacte.

On donne la condition de gain dans le cadre linéaire (en opposition à affine) :

Définition 6.26 – Dessein gagnant

| Un dessein **gagnant** est un dessein simple, présent, uniforme, obstiné et exact.

Dans le cadre affine, un dessein gagnant est simplement un dessein simple, présent, uniforme et obstiné (sans condition de parcimonie ou d'exactitude).

Proposition 6.19 – Réseau gagnant

| Un réseau de desseins gagnants normalise vers un dessein gagnant.

6.5.2 Correction

On interprète un dessein simple \mathcal{D} par son plongement en ludique probabiliste (noté \mathcal{D}_p pour une meilleure lisibilité).

Proposition 6.20 – Dessein bi-matériel

| Pour tout dessein \mathcal{D} d'un séquent de bi-comportements simple $\Xi \vdash \Lambda$, le dessein \mathcal{D}_p (= \mathcal{D} vu comme dessein probabiliste) appartient au séquent de bi-comportements probabiliste $\Xi^{\perp_p \perp_p} \vdash \Lambda^{\perp_p \perp_p}$. Le dessein \mathcal{D} est bi-matériel si et seulement si \mathcal{D}_p est bi-matériel.

Démonstration : La proposition 6.18 page 165 implique que \mathfrak{D} est uniforme si et seulement si \mathfrak{D}_p est uniforme.

Montrons que si \mathfrak{D} est bi-matériel alors \mathfrak{D}_p est bi-matériel. Le dessein \mathfrak{D} est bi-matériel si et seulement si

$$\forall \mathfrak{D}' \subseteq \mathfrak{D}, \mathfrak{D} \equiv \mathfrak{D}' \Rightarrow \mathfrak{D} = \mathfrak{D}'$$

Le dessein \mathfrak{D}_p est bi-matériel si et seulement si

$$\forall \mathfrak{D}'_p \subseteq \mathfrak{D}_p, d(\mathfrak{D}_p, \mathfrak{D}'_p) = 0 \Rightarrow \mathfrak{D}_p = \mathfrak{D}'_p$$

– Supposons \mathfrak{D} bi-matériel.

Soit $\mathfrak{D}'_p \subseteq \mathfrak{D}_p$ tel que $d(\mathfrak{D}_p, \mathfrak{D}'_p) = 0$. Le dessein \mathfrak{D}'_p est simple car ses coefficients sont inclus dans les coefficients d'un dessein simple \mathfrak{D}_p . On a donc $\mathfrak{D}_p \equiv \mathfrak{D}'_p$ et $\mathfrak{D}_p = \mathfrak{D}'_p$. Donc \mathfrak{D}_p est bi-matériel.

– La réciproque se fait de la même manière. ■

Les théorèmes 6.21 et 6.23 sont valides pour les conditions de gain [Gir01] suivantes :

- Un dessein est **gagnant** s'il est simple, uniforme, obstiné et parcimonieux (condition (trop faible) souhaitée pour MALL_2 , voir la remarque 1.8 page 48).
- Un dessein est **gagnant** s'il est simple, uniforme, obstiné et exact (condition pour MALL_2).
- Un dessein est **gagnant** s'il est simple, uniforme et obstiné (condition pour MAAL_2).

Théorème 6.21 – Correction

A tout dessein (simple) \mathfrak{D} d'un séquent de bi-comportements simple $\Xi \vdash \Lambda$, le dessein \mathfrak{D}_p (= \mathfrak{D} vu comme dessein probabiliste) appartient au séquent de bi-comportements probabiliste $\Xi^{\perp_P \perp_P} \vdash \Lambda^{\perp_P \perp_P}$. Le dessein \mathfrak{D}_p est gagnant (resp. bi-matériel) si et seulement si \mathfrak{D} est gagnant (resp. bi-matériel).

Démonstration : La conservation de la condition de gain repose uniquement sur la conservation de l'uniformité et de la bi-matérialité, les autres propriétés étant identiques.

- La proposition 6.18 page 165 stipule que \mathfrak{D} est uniforme si et seulement si \mathfrak{D}_p est uniforme.
- La proposition 6.20 page précédente implique que \mathfrak{D} est bi-matériel si et seulement si \mathfrak{D}_p est bi-matériel. ■

Soit ψ la traduction ϕ de MALL_2 (resp. MAAL_2) en ludique simple composée avec le plongement de la ludique simple en ludique probabiliste. Par composition avec le théorème de correction de la ludique simple, on obtient le théorème suivant.

Théorème 6.22 – Correction

Si π est une preuve d'une formule close A alors $\psi(\pi) \in \psi(A)$ et $\psi(\pi)$ est gagnant et bi-matériel.

6.5.3 Complétude

Théorème 6.23 – Complétude

Soit \mathcal{D} un dessein bi-matériel et gagnant d'un séquent de bi-comportements probabiliste \mathfrak{s} -régulier $\Xi \vdash \Lambda$. Alors, le dessein \mathcal{D}_s ($= \mathcal{D}$ vu comme dessein simple) appartient au séquent de bi-comportements simple $\text{Simp}(\Xi) \vdash \text{Simp}(\Lambda)$ et est bi-matériel et gagnant dans $\text{Simp}(\Xi) \vdash \text{Simp}(\Lambda)$.

Démonstration : La condition de gain sur \mathcal{D} implique que \mathcal{D} est simple et donc que \mathcal{D}_s existe bien. De plus, \mathcal{D}_s est naturellement gagnant. Le dessein \mathcal{D}_s est bi-matériel par la proposition 6.20 page 166. ■

Par composition avec le théorème de complétude de la ludique simple, on obtient le théorème suivant (pour les conditions de gain correspondant à MALL_2 et MAAL_2).

Proposition 6.24 – Complétude

Si A est une formule close Π_1 et $\mathcal{D} \in \psi(A)$ est gagnant et bi-matériel alors il existe une preuve π de A telle que $\mathcal{D} = \psi(\pi)$.

Troisième partie
Ludique avec répétitions

Chapitre 7

Ludique avec pointeurs

Sommaire

7.1	Intuitions	172
7.2	Desseins et réseaux	173
7.2.1	Desseins	173
7.2.2	Réseaux	176
7.3	Normalisation	177
7.3.1	Intuitions	177
7.3.2	Machine abstraite	178
7.3.3	Exemples de normalisations	183
7.4	Ludique avec pointeurs et arbres de Böhm abstraits	185
7.4.1	Arbres de Böhm abstraits	186
7.4.2	Validité des définitions	187
7.5	Plongement de la ludique simple	190
7.6	Ordres	190
7.7	À propos des théorèmes analytiques	191
7.7.1	Non-séparation	191
7.7.2	Associativité	193
7.7.3	Stabilité	193
7.7.4	Monotonie	194
7.8	Comportements	194
7.8.1	Définitions	195
7.8.2	Notions pour la complétude interne	197
7.9	Connecteurs	199
7.9.1	Décalages	199
7.9.2	Exponentielles	200
7.9.3	Additifs	201
7.9.4	Multiplicatifs	202

Ce chapitre décrit une extension de la ludique avec des répétitions en vue des exponentielles.

Les desseins de la ludique simple sont des arbres de Böhm abstraits [Cur98, CH98] sans répétition [Cur04b]. L'objet de ce chapitre est de rajouter des pointeurs pour retrouver les arbres de Böhm abstraits. La notion de pointeur, qui provient des jeux de Hyland et Ong [HO00], permet de représenter des répétitions de coups sur un lieu donné.

Le plan de ce chapitre suit celui du chapitre 1 jusqu'à la définition des connecteurs où des exponentielles sont définies.

Cela ne suffit pas pour le théorème de séparation. Pour corriger ce défaut, des coefficients sont rajoutés au chapitre 8 page 205. L'uniformité n'est pas non plus abordée dans ce chapitre car le manque de séparation montre un manque d'opposants.

7.1 Intuitions

La notion de pointeur est celle qui permet de rendre compte de la refocalisation. Ainsi, l'arbre suivant

$$\begin{array}{c}
 (+, \xi.0.1, \{42\}) \\
 | \\
 (-, \xi.0, \{1; 2\}) \\
 | \\
 (+, \xi, \{0; 2\}) \\
 | \\
 (-, \xi.0, \{1\}) \\
 | \\
 (+, \xi, \{0; 1\})
 \end{array}$$

n'est pas un dessin simple car le lieu ξ est utilisé deux fois. Le problème crucial à résoudre est de savoir qui justifie l'action $(+, \xi.0.1, \{42\})$. Est-ce la seconde ou la quatrième action ?

Cette question est résolue par la notion de pointeur.

L'arbre précédent doit donc s'écrire en fait :

$$\begin{array}{cc}
 (+, 0 : \xi.0.1, \{42\}) & (+, 1 : \xi.0.1, \{42\}) \\
 | & | \\
 (-, \xi.0, \{1; 2\}) & (-, \xi.0, \{1; 2\}) \\
 | & | \\
 (+, \xi, \{0; 2\}) & (+, \xi, \{0; 2\}) \\
 | & | \\
 (-, \xi.0, \{1\}) & (-, \xi.0, \{1\}) \\
 | & | \\
 (+, \xi, \{0; 1\}) & (+, \xi, \{0; 1\})
 \end{array}
 \quad \text{ou}$$

L'action $(+, 0 : \xi.0.1, \{42\})$ signifie que le justifiant est l'action $(-, \xi.0, \{1; 2\})$ tandis que l'action $(+, 1 : \xi.0.1, \{42\})$ signifie que le justifiant est l'action $(-, \xi.0, \{1\})$ (on compte le nombre d'actions négatives sur le lieu $\xi.0$ entre l'action justifiée et le justifiant).

Les autres actions n'ont pas de pointeurs " n :" pour deux raisons distinctes :

- Les actions négatives sont justifiées par les actions positives précédentes, le pointeur “ n :” est donc toujours à zéro (et n’est pas noté).
- Les actions sur le lieu ξ sont sur un lieu de la base et ne pointent donc sur aucune action.

Naturellement, la question de la normalisation est cruciale : des pointeurs différents n’entraînent pas les mêmes normalisations. Considérons les trois desseins \mathfrak{D}_1 , \mathfrak{D}_2 et \mathfrak{E} suivants :

$$\begin{array}{ccc}
 & (+, 0 : \xi.0.1, \{42\}) & (+, 1 : \xi.0.1, \{42\}) \\
 & | & | \\
 & (-, \xi.0, \{1; 2\}) & (-, \xi.0, \{1; 2\}) \\
 & | & | \\
 & (+, \xi, \{0; 2\}) & (+, \xi, \{0; 2\}) \\
 & | & | \\
 & (-, \xi.0, \{1\}) & (-, \xi.0, \{1\}) \\
 \mathfrak{D}_1 = & | & | \\
 & (+, \xi, \{0; 1\}) & (+, \xi, \{0; 1\}) \\
 & & \\
 & (+, \xi_2, \emptyset) & (+, \xi_1, \emptyset) \\
 & | & | \\
 & (-, \xi.0.1, \{42\}) & (-, \xi.0.1, \{42\}) \\
 & | & | \\
 & (+, 0 : \xi.0, \{1\}) & (+, 0 : \xi.0, \{1; 2\}) \\
 \mathfrak{E} = & | & | \\
 & (-, \xi, \{0; 1\}) & (-, \xi, \{0; 2\})
 \end{array}$$

sur les bases respectives $\vdash \xi$, $\vdash \xi$ et $\xi \vdash \xi_1, \xi_2$. Le résultat de la normalisation du réseau \mathfrak{D}_1 , \mathfrak{E} est $(+, \xi_1, \emptyset)$ alors que celui du réseau \mathfrak{D}_2 , \mathfrak{E} est $(+, \xi_2, \emptyset)$. Les détails des interactions sont décrits page 184.

7.2 Dessins et réseaux

Les desseins simples sont généralisés en autorisant des pointeurs $n : \xi$ à la place des lieux pour les actions positives.

7.2.1 Dessins

Définition 7.1 – Pointeur

Les **pointeurs** sont engendrés par la grammaire suivante :

$$\pi := \xi \mid n : \xi$$

où ξ est un lieu et n un entier dans \mathbb{N} .

Par abus de langage, l’entier n est parfois aussi appelé **pointeur**.

On notera π_ξ pour évoquer ξ ou $n : \xi$.

Définition 7.2 – Action avec pointeurs

Sur une base $\Xi \vdash \Lambda$, une **action** est :

- soit positive : une action propre de la forme $(+, \xi, I)$ ou $(+, n : \xi, I)$ ou une action impropre \mathfrak{X} , \mathfrak{X}_ξ ou $\mathfrak{X}_{n:\xi}$;
- soit négative : la divergence négative \mathfrak{Yid}^- (l'action impropre négative) ou une action propre de la forme $(-, \xi, I)$

où $n : \xi$ est un pointeur, $I \subset_{fin} \mathbb{N}$ est une ramification et ξ un lieu.

Les actions $(+, \xi, I)$ apparaissent uniquement pour $\xi \in \Lambda$.

Pour définir les desseins, il faut auparavant définir les arbres avec pointeurs.

Définition 7.3 – Arbre avec pointeurs

Un **arbre avec pointeurs** \mathcal{D} est un arbre fini ou infini engendré par la grammaire suivante :

arbre	\mathcal{D}	:=	$\mathcal{D}^+ \mid \mathcal{D}_\xi^-$
arbre positif	\mathcal{D}^+	:=	$\mathfrak{Yid} \mid \mathcal{D}_t^+$
arbre positif non vide	\mathcal{D}_t^+	:=	$\mathfrak{X} \mid \mathfrak{X}_{\pi_\xi} \mid (+, \pi_\xi, \{i_1, \dots, i_n\}).(\mathcal{D}_{\xi.i_1}^-, \dots, \mathcal{D}_{\xi.i_n}^-)$
arbre négatif	\mathcal{D}_ξ^-	:=	$\mathfrak{Yid}^- \mid \mathcal{D}_{s_\xi}^-$
arbre négatif non vide	$\mathcal{D}_{s_\xi}^-$:=	$((-, \xi, I). \mathcal{D}_t^+)_{I \in \mathbb{N}}$
pointeur	π_ξ	:=	$\xi \mid n : \xi$

où \mathbb{N} est un ensemble de sous-parties finies de \mathbb{N} ; π_ξ représente un pointeur, n est un entier et ξ un lieu.

Un arbre simple est **positif** s'il est de la forme \mathcal{D}^+ et **négatif**, s'il est de la forme \mathcal{D}_ξ^- .

Définition 7.4 – Justification

Une suite alternée d'actions est **justifiée** sur une base β si pour tout préfixe \mathbf{c} de la suite se terminant par une action K alors

- ou bien K est le démon \mathfrak{X} ou la divergence négative \mathfrak{Xi}^- ;
- ou bien K est un démon focalisé \mathfrak{X}_ξ et ξ^+ est dans la base β ;
- ou bien K est un démon focalisé $\mathfrak{X}_{n:\xi,i}$ et \mathbf{c} est de la forme

$$\mathbf{c}' \cdot (-, \xi, J) \cdot \mathbf{c}_n \cdot (-, \xi, K_n) \dots \mathbf{c}_1 \cdot (-, \xi, K_1) \mathbf{c}_0 \cdot \mathfrak{X}_{n:\xi,i}$$

avec $i \in J$ et ξ n'apparaît pas dans les lieux des actions des (\mathbf{c}_k) .

On dit que $(-, \xi, J)$ **justifie** $\mathfrak{X}_{n:\xi,i}$.

- ou bien l'action K s'écrit (ϵ, ξ, I) et ξ^ϵ est dans la base β .
- ou bien l'action K s'écrit $(+, n : \xi.i, I)$, \mathbf{c} est de la forme

$$\mathbf{c}' \cdot (-, \xi, J) \cdot \mathbf{c}_n \cdot (-, \xi, K_n) \dots \mathbf{c}_1 \cdot (-, \xi, K_1) \mathbf{c}_0 \cdot (+, n : \xi.i, I)$$

avec $i \in J$ et ξ n'apparaît pas dans les lieux des actions des (\mathbf{c}_k) .

On dit que $(-, \xi, J)$ **justifie** $(+, n : \xi.i, I)$.

- ou bien l'action K s'écrit $(-, \xi.i, I)$ et l'action précédente dans \mathbf{c} est de la forme $(+, \xi, J)$ (resp. $(+, n : \xi, J)$) avec $i \in J$.

On dit que $(+, \xi, J)$ (resp. $(+, n : \xi, J)$) **justifie** $(-, \xi.i, I)$.

Ainsi, les pointeurs $n : \xi.i$ représentent des lieux avec une information à la “indices de De Bruijn”. Cependant, le n ne compte pas les actions quelconques entre une action et son justifiant mais simplement les actions sur les lieux ξ . Ce système simplifie les définitions et la lecture du dessin.

Définition 7.5 – Pré-chronique avec pointeurs

Une **pré-chronique avec pointeurs** sur une base β est une suite d'actions avec pointeurs de polarités alternées, justifiée, sans démon ni divergence négative \mathfrak{Xi}^- à l'intérieur.

Une pré-chronique est **positive** (resp. **négative**) si elle se termine par une action positive (resp. négative).

Définition 7.6 – Chronique

Une **chronique** est

- une pré-chronique avec pointeurs positive.
- une pré-chronique avec pointeurs (négative) dont la dernière action est \mathfrak{Xi}^- .

On note Chron_β l'ensemble des chroniques sur une base β .

Définition 7.7 – Desseins

Un arbre avec pointeurs **vide** est un arbre avec pointeurs qui ne contient pas d'actions.

Un **dessein total** sur une base β est un arbre avec pointeurs différent de \mathfrak{Fid}^- , non vide si β est positive et dont les branches sont des chroniques sur β .

Le dessein partiel \mathfrak{Fid} est l'arbre vide sur une base positive.

Le dessein \mathfrak{Fid}^- est le dessein partiel négatif.

Le scons \mathfrak{Sunk} est l'arbre vide sur une base négative. C'est un dessein total.

7.2.2 Réseaux

On réutilise une fois de plus la notion de réseau de la section 1.2.3 page 30.

Définition 7.8 – Pré-réseau

Un **pré-réseau** est un ensemble fini de desseins sur des bases β_i tel qu'un lieu apparaît au plus dans deux bases : une fois positivement et une fois négativement.

Définition 7.9 – Coupure

Une **coupure** dans un pré-réseau \mathfrak{R} est la donnée de deux desseins de \mathfrak{R} dont les bases ont un lieu en commun.

Définition 7.10 – Réseau

Un **réseau** est un pré-réseau dans lequel la relation de coupure est connexe et acyclique.

Définition 7.11 – Base de réseau

La **base** d'un réseau \mathfrak{R} est la base formée de l'ensemble des lieux (avec leur polarité) des bases des desseins de \mathfrak{R} qui n'apparaissent qu'une fois.

Définition 7.12 – Réseau clos

Un réseau est **clos** lorsque sa base est vide.

Définition 7.13 – Actions et lieux visibles

Soit \mathfrak{R} un réseau.

Un lieu est **visible** dans \mathfrak{R} si c'est un sous-lieu d'un lieu de la base de \mathfrak{R} .

Une action dans un dessein de \mathfrak{R} est **visible** dans l'un des cas suivant :

- la divergence négative \mathfrak{Fid}^- à la racine du dessein principal où après une action positive visible ;
- le démon ;
- un démon focalisé dont le lieu est visible ;
- une action propre dont le lieu est visible.

7.3 Normalisation

La normalisation nécessite une machine abstraite plus évoluée que dans le cadre simple ou probabiliste.

7.3.1 Intuitions

Les intuitions derrière la normalisation sont les suivantes :

- Lors d'une interaction, la même action peut être jouée plusieurs fois ce qui implique que la même partie d'un dessin peut servir plusieurs fois.
- Pour prendre cela en compte et faire interagir les actions entre elles, on utilise une notion d'historique des interactions qui décrit précisément le parcours fait jusqu'à un certain point.
- La notion d'ordre (définition 1.28 page 33) utilisée en ludique simple n'est pas suffisamment expressive.

Reprenons les desseins

$$\begin{array}{cc}
 \begin{array}{c}
 (+, 0 : \xi.0.1, \{42\}) \\
 | \\
 (-, \xi.0, \{1; 2\}) \\
 | \\
 (+, \xi, \{0; 2\}) \\
 | \\
 (-, \xi.0, \{1\}) \\
 | \\
 (+, \xi, \{0; 1\})
 \end{array}
 &
 \begin{array}{c}
 (+, 1 : \xi.0.1, \{42\}) \\
 | \\
 (-, \xi.0, \{1; 2\}) \\
 | \\
 (+, \xi, \{0; 2\}) \\
 | \\
 (-, \xi.0, \{1\}) \\
 | \\
 (+, \xi, \{0; 1\})
 \end{array} \\
 \mathfrak{D}_1 = & \mathfrak{D}_2 =
 \end{array}$$

$$\begin{array}{cc}
 \begin{array}{c}
 (+, \xi_2, \emptyset) \\
 | \\
 (-, \xi.0.1, \{42\}) \\
 | \\
 (+, 0 : \xi.0, \{1\}) \\
 | \\
 (-, \xi, \{0; 1\})
 \end{array}
 &
 \begin{array}{c}
 (+, \xi_1, \emptyset) \\
 | \\
 (-, \xi.0.1, \{42\}) \\
 | \\
 (+, 0 : \xi.0, \{1; 2\}) \\
 | \\
 (-, \xi, \{0; 2\})
 \end{array} \\
 \mathfrak{E} = &
 \end{array}$$

La normalisation du réseau $\mathfrak{D}_1, \mathfrak{E}$ se passe ainsi :

1. la première action dans \mathfrak{D}_1 est $(+, \xi, \{0, 1\})$ et \mathfrak{E} possède une action $(-, \xi, \{0, 1\})$ donc la première étape est $(\xi, \{0, 1\})$;
2. la continuation de $(-, \xi, \{0, 1\})$ est $(+, 0 : \xi.0, \{1\})$ qui interagit avec l'action négative duale $(-, \xi.0, \{1\})$ qui suit l'action $(+, \xi, \{0, 1\})$ jouée à la première étape ;
3. l'action positive qui suit l'action $(-, \xi.0, \{1\})$ est l'action $(+, \xi, \{0; 2\})$ qui interagit avec l'action $(-, \xi, \{0; 2\})$ de \mathfrak{E} ;
4. la continuation de $(-, \xi, \{0; 2\})$ est $(+, 0 : \xi.0, \{1; 2\})$ qui interagit avec l'action $(-, \xi.0, \{1; 2\})$ qui suit l'action positive de l'étape précédente ;
5. l'action positive suivante est $(+, 0 : \xi.0.1, \{42\})$ qui interagit avec l'action négative $(-, \xi.0.1, \{42\})$ qui suit la dernière action positive (c'est l'action $(-, \xi.0.1, \{42\})$ de la branche de droite de \mathfrak{E}).

6. le résultat de la normalisation est $(+, \xi_1, \emptyset)$ qui suit l'action négative précédente car le lieu n'est pas coupé.

La normalisation du réseau $\mathfrak{D}_2, \mathfrak{E}$ est à peine différente : les deux dernières étapes sont remplacées par

5. l'action positive suivante est $(+, 1 : \xi.0.1, \{42\})$ qui interagit avec l'action négative $(-, \xi.0.1, \{42\})$ qui suit l'avant-dernière action positive qui focalise sur $\xi.0$ (c'est l'action $(-, \xi.0.1, \{42\})$ de la branche de gauche de \mathfrak{E});
6. le résultat de la normalisation est $(+, \xi_2, \emptyset)$ qui suit l'action négative précédente car le lieu n'est pas coupé.

Une version plus formelle de cet exemple est donnée section 7.3.3 page 183.

7.3.2 Machine abstraite

La machine abstraite utilisée est une adaptation de la SVAM (Strong View Abstract Machine) décrite par Pierre-Louis Curien [Cur98].

La justification de la validité des définitions données ici est apportée par codage dans les arbres de Böhm abstraits (section 7.4 page 185). Ainsi, on ne s'occupe pas de justifier les définitions dans cette section.

Une interaction est une suite de coups qu'on appelle historique en référence aux arbres de Böhm abstraits.

Définition 7.14 – Historique

Les **historiques finis** sont engendrés par la grammaire :

$$\begin{array}{lll}
 \text{historiques} & \mathcal{H} & := \mathcal{H}_e \mid \mathcal{H}_n \\
 \text{historiques extensibles} & \mathcal{H}_e & := \epsilon \mid \mathcal{H}_e.\text{mov} \\
 \text{historiques non extensibles} & \mathcal{H}_n & := \mathcal{H}_e.\text{Vis}_{(\mathfrak{Fid}^-)} \mid \mathcal{H}_e.\mathfrak{Fid} \mid \mathcal{H}_e.\mathfrak{X} \mid \mathcal{H}_e.\text{Vis}_{(\mathfrak{X}\pi)} \\
 \text{mouvements} & \text{mov} & := \text{Cut}_{\pi,I} \mid \text{Vis}_{(+,\pi,I)} \mid \text{Vis}_{(-,\xi,I)}
 \end{array}$$

où $\pi := \xi \mid n : \xi$ est un pointeur.

Quelques commentaires sur les historiques sont utiles :

- Les historiques extensibles correspondent aux interactions non forcément terminées alors que les historiques non extensibles représentent des interactions terminées.
- L'historique ϵ est l'historique vide : l'interaction n'a pas encore commencée. On n'écrit ϵ que dans cet historique. Ainsi, par exemple, l'historique $\epsilon.\text{Cut}_{\xi,I}.\text{Cut}_{\xi,i,J}$ sera noté $\text{Cut}_{\xi,I}.\text{Cut}_{\xi,i,J}$.
- L'historique $\mathcal{H}.\text{Vis}_{(\mathfrak{Fid}^-)}$ correspond à une action \mathfrak{Fid}^- visible.
- L'historique $\mathcal{H}.\mathfrak{Fid}$ correspond à la divergence en temps finie.
- L'historique $\mathcal{H}.\mathfrak{X}$ correspond au lancement d'un démon (éventuellement focalisé si le dessein négatif correspondant est différent de \mathfrak{Fid}^-).
- L'historique $\text{Cut}_{\pi,I}$ correspond à un couple d'actions duales jouées simultanément.
- L'historique $\mathcal{H}.\text{Vis}_{(\mathfrak{X}\pi)}$ correspond à un démon focalisé visible (le lieu de focalisation n'est pas coupé).
- L'historique $\text{Vis}_{(+,\pi,I)}$ correspond à une action positive propre visible (le lieu de focalisation n'est pas coupé).

- L'historique $\text{Vis}_{(-,\xi,I)}$ correspond à une action négative propre visible (le lieu de focalisation n'est pas coupé).

La notion de saut est une notion partielle. Elle permet de définir l'état de la mémoire au moment où le justifiant d'une action a été joué. On note \perp une position impossible.

Définition 7.15 – Saut

La fonction de **saut** $(n, \xi) \mapsto \text{Saut}_{\mathcal{H}}(n, \xi)$ dans un historique extensible \mathcal{H} est une fonction partielle définie par :

$$\text{Saut}_{\mathcal{H}. \text{Cut}_{k:\xi',I}}(n, \xi) = \begin{cases} \mathcal{H}. \text{Cut}_{k:\xi',I} & \text{si } n = 0 \text{ et } \exists i, \xi = \xi'.i \\ \text{Saut}_{\text{Elague}(\text{Saut}_{\mathcal{H}}(k,\xi'))}((n-1), \xi) & \text{si } n > 0 \text{ et } \exists i, \xi = \xi'.i \\ \text{Saut}_{\text{Elague}(\text{Saut}_{\mathcal{H}}(k,\xi'))}(n, \xi) & \text{sinon} \end{cases}$$

$$\text{Saut}_{\mathcal{H}. \text{Cut}_{\xi',I}}(n, \xi) = \begin{cases} \mathcal{H}. \text{Cut}_{\xi',I} & \text{si } n = 0 \text{ et } \exists i, \xi = \xi'.i \\ \perp & \text{sinon} \end{cases}$$

$$\text{Saut}_{\mathcal{H}. \text{Vis}_{(+,k:\xi',J)}. \text{Vis}_{(-,\xi'.j,I)}}(n, \xi) = \begin{cases} \mathcal{H}. \text{Vis}_{(+,k:\xi',J)}. \text{Vis}_{(-,\xi'.j,I)} & \text{si } n = 0 \text{ et } \exists i, \xi = \xi'.j.i \\ \text{Saut}_{\mathcal{H}}((n-1), \xi) & \text{si } n > 0 \text{ et } \exists i, \xi = \xi'.j.i \\ \text{Saut}_{\mathcal{H}}(n, \xi) & \text{sinon} \end{cases}$$

$$\text{Saut}_{\mathcal{H}. \text{Vis}_{(+,\xi',J)}. \text{Vis}_{(-,\xi'.j,I)}}(n, \xi) = \begin{cases} \mathcal{H}. \text{Vis}_{(+,\xi',J)}. \text{Vis}_{(-,\xi'.j,I)} & \text{si } n = 0 \text{ et } \exists i, \xi = \xi'.j.i \\ \text{Saut}_{\mathcal{H}}((n-1), \xi) & \text{si } n > 0 \text{ et } \exists i, \xi = \xi'.j.i \\ \text{Saut}_{\mathcal{H}}(n, \xi) & \text{sinon} \end{cases}$$

$$\text{Saut}_{\mathcal{H}}(n, \xi) = \perp \text{ dans les autres cas}$$

où $\text{Elague}(\)$ est la fonction qui enlève le dernier élément à une suite.

Cette notion est partielle car on ne suppose rien sur les historiques. Elle est totale sur les historiques définis aux définitions 7.16 et 7.17 (c'est une conséquence du codage dans les arbres de Böhm abstraits).

Plutôt que de définir une notion de vue comme c'est habituel en sémantique de jeux, la technique utilisée ici consiste à associer un ensemble d'actions à un historique qui correspondent aux vues joueur et opposant (les actions associées sont au-dessus de chroniques qui correspondent à la notion de vue en ludique [FH02]).

À un historique \mathcal{H} , on associe un **ensemble d'actions** $\mathcal{A}_{\mathcal{H}}$ de telle sorte :

- Si \mathcal{H} termine par $\text{Cut}_{\pi_{\xi},I}$ alors on associe une action propre négative $(-, \xi, I)$ et une action propre positive $(+, \pi_{\xi}, I)$.
- Si \mathcal{H} termine par $\text{Vis}_{(+,\pi,I)}$ alors on associe une action propre positive $(+, \pi, I)$.
- Si \mathcal{H} termine par $\text{Vis}_{(-,\xi,I)}$ alors on associe une action négative $(-, \xi, I)$.

- Si \mathcal{H} termine par le démon alors associe un démon (focalisé ou non).
- Si \mathcal{H} termine par $\text{Vis}_{(\mathfrak{F}\text{id}^-)}$ alors associe une action $\mathfrak{F}\text{id}^-$.
- Sinon, $\mathcal{A}_{\mathcal{H}}$ est vide.

Définition 7.16 – Actions associées à un historique

On associe un **ensemble d'actions** $\mathcal{A}_{\mathcal{H}}$ à un historique \mathcal{H} d'un réseau \mathfrak{R} . On procède par cas de la manière suivante :

1. Si $\mathcal{H} = \mathcal{H}' \cdot \mathfrak{F}\text{id}$ alors $\mathcal{A}_{\mathcal{H}} = \emptyset$.
2. \mathcal{H} est de longueur inférieure ou égale à 1 :
 - Si $\mathcal{H} = \epsilon$ alors $\mathcal{A}_{\mathcal{H}} = \emptyset$.
 - Si $\mathcal{H} = \mathfrak{F}\text{id}^-$ alors $\mathcal{A}_{\mathcal{H}} = \{\sigma\}$ où σ est l'action $\mathfrak{F}\text{id}^-$ à la racine du dessein principal.
 - Si $\mathcal{H} = \mathfrak{X}$ alors $\mathcal{A}_{\mathcal{H}} = \{\sigma\}$ où σ est le démon (focalisé ou non) à la racine du dessein principal.
 - Si $\mathcal{H} = \text{Cut}_{\xi, I}$ alors $\mathcal{A}_{\mathcal{H}} = \{\sigma, \tau\}$ où σ est l'action $(+, \xi, I)$ à la racine du dessein principal de \mathfrak{R} et τ est l'unique action négative duale (à la racine du dessein négatif de base $\xi \vdash \Lambda$).
 - Si $\mathcal{H} = \text{Vis}_{(-, \xi, I)}$ alors $\mathcal{A}_{\mathcal{H}} = \{\sigma\}$ où σ est l'action $(-, \xi, I)$ à la racine du dessein principal.
 - Si $\mathcal{H} = \text{Vis}_{(+, \xi, I)}$ alors $\mathcal{A}_{\mathcal{H}} = \{\sigma\}$ où σ est l'action $(+, \xi, I)$ à la racine du dessein principal.
3. \mathcal{H} est de longueur supérieure ou égale à 2 :
 - Si $\mathcal{H} = \mathcal{H}' \cdot \mathfrak{F}\text{id}^-$ alors $\mathcal{A}_{\mathcal{H}} = \emptyset$.
 - Si $\mathcal{H} = \mathcal{H}' \cdot \mathfrak{X}$ alors $\mathcal{A}_{\mathcal{H}} = \{\sigma\}$ où σ est le démon (focalisé ou non) au-dessus de l'action négative de $\mathcal{A}_{\mathcal{H}'}$.
 - Si $\mathcal{H} = \mathcal{H}' \cdot \text{Cut}_{\xi, I}$ alors $\mathcal{A}_{\mathcal{H}} = \{\sigma, \tau\}$ où σ est l'action $(+, \xi, I)$ au-dessus de l'action négative de $\mathcal{A}_{\mathcal{H}'}$ et τ est l'unique action négative duale (à la racine du dessein négatif de base $\xi \vdash \Lambda$).
 - Si $\mathcal{H} = \mathcal{H}' \cdot \text{Cut}_{n, \xi, I}$ alors $\mathcal{A}_{\mathcal{H}} = \{\sigma, \tau\}$ où σ est l'action $(+, n : \xi, I)$ au-dessus de l'action négative de $\mathcal{A}_{\mathcal{H}'}$ et τ est l'action négative duale au-dessus de l'action positive de $\mathcal{A}_{\text{Saut}_{\mathcal{H}'(n, \xi)}}$.
 - Si $\mathcal{H} = \mathcal{H}' \cdot \text{Vis}_{(+, \pi, I)}$ alors $\mathcal{A}_{\mathcal{H}} = \{\sigma\}$ où σ est l'action $(+, \pi, I)$ au-dessus de l'action négative de $\mathcal{A}_{\mathcal{H}'}$.
 - Si $\mathcal{H} = \mathcal{H}' \cdot \text{Vis}_{(\mathfrak{X}_{\xi})}$ alors $\mathcal{A}_{\mathcal{H}} = \{\sigma\}$ où σ est l'action \mathfrak{X}_{ξ} au-dessus de l'action négative de $\mathcal{A}_{\mathcal{H}'}$.
 - Si $\mathcal{H} = \mathcal{H}' \cdot \text{Vis}_{(-, \xi, I)}$ alors $\mathcal{A}_{\mathcal{H}} = \{\sigma\}$ où σ est l'action $(-, \xi, I)$ au-dessus de l'action positive de $\mathcal{A}_{\mathcal{H}'}$.

Définition 7.17 – Machine abstraite

Soit \mathfrak{R} un réseau. À partir de $\mathcal{H}_0 = \epsilon$, on définit une **réduction pas à pas** (non déterministe) sur les historiques. Un historique \mathcal{H} est un **historique sur le réseau** \mathfrak{R} si $\mathcal{H}_0 \rightsquigarrow^+ \mathcal{H}$.

On procède par cas.

1. $\mathcal{H} = \epsilon$. C'est le point de départ.

(a) Si le dessein principal \mathcal{D} de \mathfrak{R} est positif alors on procède par cas sur \mathcal{D} :

- si $\mathcal{D} = \mathfrak{Fid}$ alors $\mathcal{H} \rightsquigarrow \mathfrak{Fid}$;
- si $\mathcal{D} = \mathfrak{X}_\xi$ alors
 - si ξ est visible alors $\mathcal{H} \rightsquigarrow \text{Vis}_{(\mathfrak{X}_\xi)}$
 - sinon
 - si $\mathcal{D}_\xi^- = \mathfrak{Fid}^-$ alors $\mathcal{H} \rightsquigarrow \mathfrak{Fid}$
 - sinon $\mathcal{H} \rightsquigarrow \mathfrak{X}$
- si $\mathcal{D} = (+, \xi, I).(\mathcal{D}_1, \dots, \mathcal{D}_n)$ alors
 - si ξ est visible alors $\mathcal{H} \rightsquigarrow \text{Vis}_{(+, \xi, I)}$
 - sinon
 - si $(-, \xi, I)$ apparaît dans \mathcal{D}_ξ^- alors $\mathcal{H} \rightsquigarrow \text{Cut}_{\xi, I}$;
 - sinon $\mathcal{H} \rightsquigarrow \mathfrak{Fid}$;

(b) Si le dessein principal \mathcal{D} de \mathfrak{R} est négatif alors le lieu négatif ξ de la base de \mathcal{D} est visible. On procède par cas sur \mathcal{D} :

- si $\mathcal{D} = \mathfrak{Fid}^-$ alors $\mathcal{H} \rightsquigarrow \mathfrak{Fid}^-$.
- si une action négative $(-, \xi, I)$ appartient à \mathcal{D} , alors $\mathcal{H} \rightsquigarrow \mathcal{H}.\text{Vis}_{(-, \xi, I)}$.
On remarque le non-déterminisme de cette règle.
- sinon $\mathcal{D} = \mathfrak{Gfun}$ et \mathcal{H} est une forme normale.

2. $\mathcal{H} = \mathcal{H}'.\text{Cut}_{\pi, I}$ ou $\mathcal{H} = \mathcal{H}'.\text{Vis}_{(-, \xi, I)}$.

Soit σ l'action négative associée à \mathcal{H} . Soit K l'action positive suivant σ . On procède par cas suivant K .

(a) $K = (+, \xi, I)$

On procède comme pour le cas 1(a) :

- si ξ est visible alors $\mathcal{H} \rightsquigarrow \mathcal{H}.\text{Vis}_{(+, \xi, I)}$
- sinon
 - si $(-, \xi, I)$ apparaît dans \mathcal{D}_ξ^- alors $\mathcal{H} \rightsquigarrow \mathcal{H}.\text{Cut}_{\xi, I}$;
 - sinon $\mathcal{H} \rightsquigarrow \mathcal{H}.\mathfrak{Fid}$;

(b) $K = (+, n : \xi, I)$

- si ξ est visible alors $\mathcal{H} \rightsquigarrow \mathcal{H}.\text{Vis}_{(+, n : \xi, I)}$
- sinon, soit σ l'action positive de $\mathcal{A}_{\text{Saut}_{\mathcal{H}'(n, \xi)}}$. On procède par cas suivant la continuation de σ .
 - i. si $(-, \xi, I)$ suit σ alors $\mathcal{H} \rightsquigarrow \mathcal{H}.\text{Cut}_{n : \xi, I}$;
 - ii. sinon $\mathcal{H} \rightsquigarrow \mathcal{H}.\mathfrak{Fid}$.

(c) $K = \mathfrak{X}$.

On a $\mathcal{H} \rightsquigarrow \mathcal{H}.\mathfrak{X}$.

(d) $K = \mathfrak{X}_\xi$.

On procède comme pour le cas $(+, \xi, I)$:

- si ξ est visible alors $\mathcal{H} \rightsquigarrow \mathcal{H}.\text{Vis}_{(\mathfrak{X}_\xi)}$;
- sinon
 - si $\mathcal{D}_\xi^- = \mathfrak{Fid}^-$ alors $\mathcal{H} \rightsquigarrow \mathcal{H}.\mathfrak{Fid}$;
 - sinon $\mathcal{H} \rightsquigarrow \mathcal{H}.\mathfrak{X}$;

(e) $K = \mathfrak{X}_{n;\xi}$.

On procède comme pour le cas $(+, n : \xi, I)$:

– si ξ est visible alors $\mathcal{H} \rightsquigarrow \mathcal{H}.\text{Vis}_{(\mathfrak{X}_{n;\xi})}$;

– sinon, soit σ l'action positive de $\mathcal{A}_{\text{Saut}_{\mathcal{H}'(n,\xi)}}$. On procède par cas suivant la continuation \mathfrak{D}^- de σ .

– si $\mathfrak{D}^- = \mathfrak{Fid}^-$ alors $\mathcal{H} \rightsquigarrow \mathcal{H}.\mathfrak{Fid}$;

– sinon $\mathcal{H} \rightsquigarrow \mathcal{H}.\mathfrak{X}$;

3. $\mathcal{H} = \mathcal{H}'.\text{Vis}_{(+,\pi,I)}$.

Soit σ l'action (positive) associée à \mathcal{H} .

(a) Si σ est suivie d'au moins une action négative $(-, \xi.i, J)$, alors $\mathcal{H} \rightsquigarrow \mathcal{H}.\text{Vis}_{(-,\xi.i,J)}$.

On remarque le non-déterminisme de cette règle.

(b) Si σ est suivie de \mathfrak{Fid}^- alors $\mathcal{H} \rightsquigarrow \mathcal{H}.\text{Vis}_{(\mathfrak{Fid}^-)}$.

(c) Si σ n'est suivie d'aucune action négative, alors \mathcal{H} est une forme normale.

Remarque 7.1 Il y a deux règles non-déterministes qui correspondent toutes les deux au choix d'une action négative visible (à la racine du dessein principal ou après une action positive visible). Cela correspond à la duplication du jeton dans la machine LAM [Fag02].

Définition 7.18 – Oubli d'un historique

L'oubli $\mathfrak{c}_{\mathcal{H}}$ d'un historique \mathcal{H} est une pré-chronique (éventuellement vide) définie inductivement.

Le cas de $\mathfrak{c}_{\mathcal{H}.Vis_{(+,n:\xi,i,J)}}$ est traité ensuite.

$$\begin{aligned}
 \mathfrak{c}_{\epsilon} &= \epsilon \text{ (la pré-chronique vide)} \\
 \mathfrak{c}_{\mathcal{H}.\mathfrak{X}} &= \mathfrak{c}_{\mathcal{H}}.\mathfrak{X} \\
 \mathfrak{c}_{\mathcal{H}.\mathfrak{F}id} &= \mathfrak{c}'_{\mathcal{H}} \\
 \mathfrak{c}_{\mathcal{H}.Vis_{(\eta,\xi,I)}} &= \mathfrak{c}_{\mathcal{H}}.(\eta, \xi, I) \\
 \mathfrak{c}_{\mathcal{H}.Vis_{(\mathfrak{X}_{\xi})}} &= \mathfrak{c}_{\mathcal{H}}.\mathfrak{X}_{\xi} \\
 \mathfrak{c}_{\mathcal{H}.Vis_{(\mathfrak{F}id^{-})}} &= \mathfrak{c}_{\mathcal{H}}.\mathfrak{F}id^{-} \\
 \mathfrak{c}_{\mathcal{H}.Cut_{\pi,I}} &= \mathfrak{c}_{\mathcal{H}}
 \end{aligned}$$

où η est égal à $+$ ou $-$ et la pré-chronique $\mathfrak{c}'_{\mathcal{H}}$ est égale à la pré-chronique $\mathfrak{c}_{\mathcal{H}}$ privée de l'action négative (propre) maximale éventuelle.

Pour le cas $\mathfrak{c}_{\mathcal{H}.Vis_{(+,n:\xi,i,J)}}$, on doit redéfinir le pointeur $n : \xi$:

$$\mathfrak{c}_{\mathcal{H}.Vis_{(+,n:\xi,i,J)}} = \mathfrak{c}_{\mathcal{H}}.(+, m : \xi, i, J)$$

où m est le nombre d'actions de la forme $Vis_{(-,\xi,K)}$ dans $\mathfrak{c}_{\mathcal{H}}$ privé de $Saut_{\mathcal{H}}(n, \xi, i)$.

De même, pour le cas $\mathfrak{c}_{\mathcal{H}.Vis_{(\mathfrak{X}_{n:\xi,i})}}$, on doit redéfinir le pointeur $n : \xi$:

$$\mathfrak{c}_{\mathcal{H}.Vis_{(\mathfrak{X}_{n:\xi,i})}} = \mathfrak{c}_{\mathcal{H}}.(\mathfrak{X}_{m:\xi,i})$$

où m est le nombre d'actions de la forme $Vis_{(-,\xi,K)}$ dans $\mathfrak{c}_{\mathcal{H}}$ privé de $Saut_{\mathcal{H}}(n, \xi, i)$.

Proposition 7.1 – Oublis et chroniques

L'oubli d'un historique d'un réseau sur une base β est une pré-chronique (éventuellement vide) sur β .

Définition 7.19 – Normalisation

La **normalisation** $\llbracket \mathfrak{R} \rrbracket$ d'un réseau \mathfrak{R} de base β est le dessein de base β formé de l'union des oublis des historiques sur le réseau \mathfrak{R} .

Définition 7.20 – Dessesins orthogonaux

Deux desseins \mathfrak{D} et \mathfrak{E} de bases duales sont **orthogonaux** si $\llbracket \mathfrak{D}, \mathfrak{E} \rrbracket = \mathfrak{X}$.

On note cette situation $\mathfrak{D} \perp \mathfrak{E}$.

7.3.3 Exemples de normalisations

Reprenons les desseins

$$\begin{array}{ccc}
& (+, 0 : \xi.0.1, \{42\}) & (+, 1 : \xi.0.1, \{42\}) \\
& \downarrow & \downarrow \\
& (-, \xi.0, \{1; 2\}) & (-, \xi.0, \{1; 2\}) \\
& \downarrow & \downarrow \\
& (+, \xi, \{0; 2\}) & (+, \xi, \{0; 2\}) \\
& \downarrow & \downarrow \\
& (-, \xi.0, \{1\}) & (-, \xi.0, \{1\}) \\
& \downarrow & \downarrow \\
\mathfrak{D}_1 = & (+, \xi, \{0; 1\}) & \mathfrak{D}_2 = (+, \xi, \{0; 1\}) \\
& & \\
& (+, \xi_2, \emptyset) & (+, \xi_1, \emptyset) \\
& \downarrow & \downarrow \\
& (-, \xi.0.1, \{42\}) & (-, \xi.0.1, \{42\}) \\
& \downarrow & \downarrow \\
& (+, 0 : \xi.0, \{1\}) & (+, 0 : \xi.0, \{1; 2\}) \\
& \downarrow & \downarrow \\
\mathfrak{E} = & (-, \xi, \{0; 1\}) & (-, \xi, \{0; 2\})
\end{array}$$

Les historiques de la réduction pas à pas du réseau $\mathfrak{D}_1, \mathfrak{E}$ sont

$$\begin{aligned}
\epsilon &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \\
&\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0; \xi.0, \{1\}} \\
&\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0; \xi.0, \{1\}} \cdot \text{Cut}_{\xi, \{0; 2\}} \\
&\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0; \xi.0, \{1\}} \cdot \text{Cut}_{\xi, \{0; 2\}} \cdot \text{Cut}_{0; \xi.0, \{1; 2\}} \\
&\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0; \xi.0, \{1\}} \cdot \text{Cut}_{\xi, \{0; 2\}} \cdot \text{Cut}_{0; \xi.0, \{1; 2\}} \cdot \text{Cut}_{0; \xi.0.1, \{42\}} \\
&\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0; \xi.0, \{1\}} \cdot \text{Cut}_{\xi, \{0; 2\}} \cdot \text{Cut}_{0; \xi.0, \{1; 2\}} \cdot \text{Cut}_{0; \xi.0.1, \{42\}} \cdot \text{Vis}_{(+, \xi_1, \emptyset)}
\end{aligned}$$

ce qui donne

$$[[\mathfrak{D}_1, \mathfrak{E}]] = (+, \xi_1, \emptyset)$$

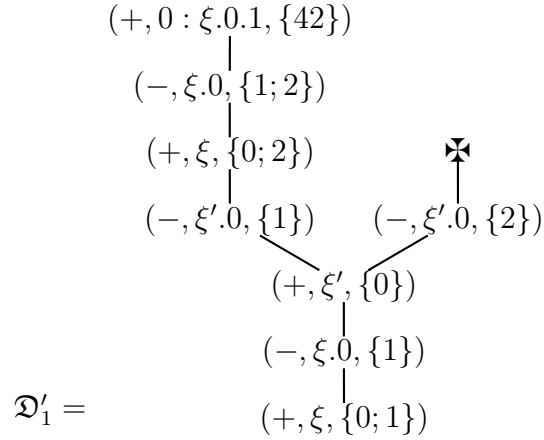
Les historiques de la réduction pas à pas du réseau $\mathfrak{D}_2, \mathfrak{E}$ sont

$$\begin{aligned}
\epsilon &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \\
&\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0; \xi.0, \{1\}} \\
&\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0; \xi.0, \{1\}} \cdot \text{Cut}_{\xi, \{0; 2\}} \\
&\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0; \xi.0, \{1\}} \cdot \text{Cut}_{\xi, \{0; 2\}} \cdot \text{Cut}_{0; \xi.0, \{1; 2\}} \\
&\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0; \xi.0, \{1\}} \cdot \text{Cut}_{\xi, \{0; 2\}} \cdot \text{Cut}_{0; \xi.0, \{1; 2\}} \cdot \text{Cut}_{1; \xi.0.1, \{42\}} \\
&\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0; \xi.0, \{1\}} \cdot \text{Cut}_{\xi, \{0; 2\}} \cdot \text{Cut}_{0; \xi.0, \{1; 2\}} \cdot \text{Cut}_{1; \xi.0.1, \{42\}} \cdot \text{Vis}_{(+, \xi_2, \emptyset)}
\end{aligned}$$

ce qui donne

$$[[\mathfrak{D}_2, \mathfrak{E}]] = (+, \xi_2, \emptyset)$$

Dans ces deux exemples, le calcul des historiques est déterministe car il n'y a pas d'actions négatives propres visibles. Pour un exemple d'utilisation de non-déterminisme dans le calcul des historiques (et non dans la normalisation), on peut considérer le dessin



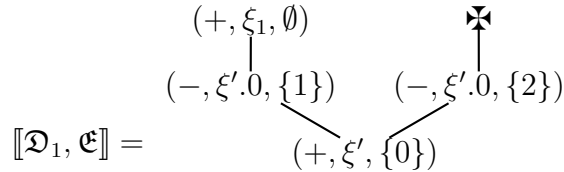
Les suites maximales d'historiques du réseau $\mathfrak{D}'_1, \mathfrak{E}$ sont alors

$$\begin{aligned}
 \epsilon &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \\
 &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0: \xi.0, \{1\}} \\
 &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0: \xi.0, \{1\}} \cdot \text{Vis}_{(+, \xi', \{0\})} \\
 &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0: \xi.0, \{1\}} \cdot \text{Vis}_{(+, \xi', \{0\})} \cdot \text{Vis}_{(-, \xi'.0, \{1\})} \\
 &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0: \xi.0, \{1\}} \cdot \text{Vis}_{(+, \xi', \{0\})} \cdot \text{Vis}_{(-, \xi'.0, \{1\})} \cdot \text{Cut}_{\xi, \{0; 2\}} \\
 &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0: \xi.0, \{1\}} \cdot \text{Vis}_{(+, \xi', \{0\})} \cdot \text{Vis}_{(-, \xi'.0, \{1\})} \cdot \text{Cut}_{\xi, \{0; 2\}} \cdot \text{Cut}_{0: \xi.0, \{1; 2\}} \\
 &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0: \xi.0, \{1\}} \cdot \text{Vis}_{(+, \xi', \{0\})} \cdot \text{Vis}_{(-, \xi'.0, \{1\})} \cdot \text{Cut}_{\xi, \{0; 2\}} \cdot \text{Cut}_{0: \xi.0, \{1; 2\}} \cdot \text{Cut}_{0: \xi.0.1, \{42\}} \\
 &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0: \xi.0, \{1\}} \cdot \text{Vis}_{(+, \xi', \{0\})} \cdot \text{Vis}_{(-, \xi'.0, \{1\})} \cdot \text{Cut}_{\xi, \{0; 2\}} \cdot \text{Cut}_{0: \xi.0, \{1; 2\}} \cdot \text{Cut}_{0: \xi.0.1, \{42\}} \cdot \\
 &\quad \quad \quad \text{Vis}_{(+, \xi_1, \emptyset)}
 \end{aligned}$$

et

$$\begin{aligned}
 \epsilon &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \\
 &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0: \xi.0, \{1\}} \\
 &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0: \xi.0, \{1\}} \cdot \text{Vis}_{(+, \xi', \{0\})} \\
 &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0: \xi.0, \{1\}} \cdot \text{Vis}_{(+, \xi', \{0\})} \cdot \text{Vis}_{(-, \xi'.0, \{2\})} \\
 &\rightsquigarrow \text{Cut}_{\xi, \{0; 1\}} \cdot \text{Cut}_{0: \xi.0, \{1\}} \cdot \text{Vis}_{(+, \xi', \{0\})} \cdot \text{Vis}_{(-, \xi'.0, \{2\})} \cdot \text{✂}
 \end{aligned}$$

ce qui donne



7.4 Ludique avec pointeurs et arbres de Böhm abstraits

L'objectif de cette section est de justifier la validité des définitions et de la proposition de la section précédente.

On rappelle les définitions et propositions sur les arbres de Böhm abstraits [Cur98]. Pour faciliter la lecture et la compréhension, les notations utilisées pour les arbres de

Böhm abstraits sont intermédiaires entre l'article de Pierre-Louis Curien et les notations de la section précédente.

7.4.1 Arbres de Böhm abstraits

Définition 7.21 – Position

Étant donné deux ensembles de coups A et X ,

- Un **coup opposant** c_o est un élément de $\{\bullet\} \cup A \cup X$.
- Un **coup joueur** c_p est un élément de X (noté $[x]$) ou une paire $[i : a]$ où $i \in \mathbb{N}$ et $a \in A$. L'entier i est appelé un **pointeur**.
- Une **position paire** est une suite non vide alternante de la forme

$$c_1^o c_2^p \dots c_{2n-1}^o c_{2n}^p$$

telle que

- $c_1^o \in \{\bullet\} \cup X$ et $\forall i > 1, c_i^o \in A$
- Si $c_{2i}^p = [k : a]$ alors $k < i$. Dans ce cas, on dit que c_{2i}^p est lié par $c_{2(i-k)-1}^o$.

Définition 7.22 – Arbre de Böhm abstrait

Un **arbre de Böhm abstrait** est un ensemble \mathcal{D} non vide clos par préfixe paire de positions commençant par \bullet telles que

$$qc^p \in \mathcal{D} \wedge qc^{p'} \in \mathcal{D} \Rightarrow c^p = c^{p'}$$

Un **renommage** sur les arbres de Böhm abstraits est défini par :

$$[x \leftarrow \phi] = \{x.r \mid \bullet.r \in \phi\}$$

Un **multi-arbre de Böhm abstrait** est une union sur les (x_i) distincts de $x_i \leftarrow \phi_i$.

Dans la définition précédente, un renommage doit être compris comme un décalage d'un dessein positif et un multi-arbre de Böhm abstrait comme un dessein négatif. Les ϕ_i intervenant dans la définition de ce dessein négatif correspondent aux desseins positifs suivant les actions négatives dans un dessein négatif total.

Définition 7.23 – Historique pour les arbres de Böhm abstraits

Un **historique** est un élément de la grammaire

$$\Gamma := \bullet_\nu \mid \Gamma.\text{Vis}_{a_\nu} \mid \Gamma.x \mid \Gamma.\langle m : a \rangle \mid \Gamma.\text{Vis}_{[\nu:a]} \mid \Gamma.\text{Vis}_{[x]}$$

Le $i^{\text{ème}}$ élément de Γ est noté $\Gamma^{(i)}$.

Un historique $\Gamma.x$ correspond à action négative coupée alors que $\Gamma.\langle m : a \rangle$ correspond à une action positive (avec un pointeur) coupée.

Définition 7.24 – Vue

La **vue** $\text{Vue}_\Gamma(n)$ d'une suite de coups est définie inductivement par :

$$\frac{}{\text{Vue}_\Gamma(1) = \bullet} \quad \frac{\Gamma^{(2n)} = x}{\text{Vue}_\Gamma(2n) = x} \quad \frac{\Gamma^{(n)} = \langle m : a \rangle}{\text{Vue}_\Gamma(n) = \text{Vue}_\Gamma(m-1).a}$$

$$\frac{\Gamma^{(n)} = \text{Vis}_{a\nu}}{\text{Vue}_\Gamma(n) = \text{Vue}_\Gamma(n-2).a}$$

7.4.2 Validité des définitions

La validité des définitions de la ludique avec pointeurs est montrée en plusieurs points :

1. Les pointeurs $n : \xi$ signifient que le justifiant d'une action $(+, n : \xi.i, I)$ est la $n^{\text{ième}}$ action de la forme $(-, \xi, J)$ dans la chronique à partir de $(+, n : \xi.i, I)$ (définition 7.4 page 175).

Il est équivalent de compter toutes les actions négatives et non simplement les actions de la forme $(-, \xi, J)$ et adapter les définitions de saut (définition 7.15 page 179) et d'oubli d'un historique (définition 7.18 page 183).

De même, quitte à adapter les définitions, il est équivalent de compter les actions depuis le début des chroniques et historiques (méthode dite de De Bruijn) ou depuis la fin (méthode dite de De Bruijn inverse).

2. Dans l'oubli d'un historique (définition 7.18 page 183), un mouvement $\text{Vis}_{(+,n:\xi.i,J)}$ est traduit en une action $(+, m : \xi.i, J)$ où m compte les actions de la forme $\text{Vis}_{(-,\xi,K)}$.

Il est équivalent de faire cet oubli directement au niveau des historiques (dans la machine abstraite, on n'empile plus $\text{Vis}_{(+,n:\xi.i,J)}$ mais $\text{Vis}_{(+,m:\xi.i,J)}$). Cela change en conséquence la notion de saut (définition 7.15 page 179).

3. Les historiques en ludique probabiliste utilisent des “actions doubles” $\text{Cut}_{n:\xi,I}$ alors que les arbres de Böhm abstraits décomposent en deux $\langle n : \xi, I \rangle$ et $[\xi, I]$ (dans cet ordre).
4. Les historiques des arbres de Böhm abstraits utilisent des variables ν comme expliqué dans l'article de Pierre-Louis Curien [Cur98] pour calculer les justifiants dans l'oubli des historiques avec une fonction (readback) appelée à la fin. Ces variables ne sont pas indispensables : il suffit de reconstruire la vue comme on le fait à la définition 7.18 page 183.

La définition suivante montre une traduction des desseins dans une famille d'arbres de Böhm abstraits (c'est un famille car il faut choisir les ensembles de coups A et le X pour la définition 7.22 page ci-contre).

Définition 7.25 – Traduction des desseins dans les arbres de Böhm abstraits

Soit $A = \{(\xi, I); \Omega; \mathfrak{X}_\xi\}$ et $X = \{\mathfrak{X}\}$. Les desseins et les arbres de Böhm abstraits étant des arbres, la traduction s'occupe (presque uniquement) de traduire les actions en coups.

- le dessein \mathfrak{Fid} est traduit en Ω ;
- un dessein négatif différent de \mathfrak{Fid}^- sur un lieu ξ est traduit par $[\mathfrak{X}_\xi.\mathfrak{X}]$ et la traduction des autres actions négatives ;
- une action \mathfrak{X} est traduite en $[\mathfrak{X}]$;
- une action \mathfrak{X}_ξ est traduite en $[\mathfrak{X}_\xi]$;
- une action $(+, \xi, I)$ est traduite en $[\xi, I]$;
- une action $(-, \xi, I)$ est traduite en $[\xi, I]$;
- une action $(+, n : \xi.i, I)$ est traduite en $[\bar{n} : \xi.i, I]$ où \bar{n} est le nombre d'actions négatives entre son justifiant et le début de la chronique et non simplement le nombre d'actions négatives de lieu ξ entre $(+, n : \xi.i, I)$ et son justifiant ;
- une action $\mathfrak{X}_{n:\xi.i}$ est traduite en $[\bar{n} : \mathfrak{X}_{\xi.i}]$ où \bar{n} est le nombre d'actions négatives entre son justifiant et le début de la chronique et non simplement le nombre d'actions négatives de lieu ξ entre $\mathfrak{X}_{n:\xi.i}$ et son justifiant.

La justification suit celle de la ludique avec pointeurs.

Remarque 7.2 La traduction d'un dessein négatif différent de \mathfrak{Fid}^- utilise une sorte d'action duale de \mathfrak{X}_ξ suivie d'un démon ce qui permet de voir \mathfrak{X}_ξ comme une action normale.

Le traduit d'un dessein positif est un arbre de Böhm abstrait. Le traduit d'un dessein négatif est un multi-arbre de Böhm abstrait. Le traduit d'un réseau est un "réseau" d'arbres de Böhm abstraits.

La traduction inverse donne un algorithme pour passer du résultat d'une normalisation d'arbres de Böhm abstraits à un dessein.

Définition 7.26 – Traduction inverse

On utilise la convention de notation stipulant que $(-, \xi, I). \mathfrak{Fid}$ signifie que $(-, \xi, I)$ n'apparaît pas dans le dessein à cette position.

Les actions propres sont traduites facilement :

- un coup joueur $[\xi, I]$ est traduit en $(+, \xi, I)$;
- un coup opposant $[\xi, I]$ est traduit en $(-, \xi, I)$;
- un coup $[n : \xi, I]$ est traduit en $(+, \bar{n} : \xi, I)$ où \bar{n} est la transformation inverse de la précédente.

Pour les actions impropres, il faut tenir compte des démons et des divergences :

- le dessein \bullet sur une base positive est traduit en \mathfrak{Fid} ;
- le dessein Ω sur une base positive est traduit en \mathfrak{Fid} ;
- un coup $[\boxtimes]$ est traduit en \boxtimes ;
- un coup $[n : \boxtimes_\xi]$ est traduit en $\boxtimes_{\bar{n}:\xi}$.
- un coup $[\boxtimes_\xi]$ positif est traduit en \boxtimes_ξ ;
- une position négative sans coups $[\xi, I]$ est traduite par :
 - le sconse $\mathfrak{S}\mathfrak{u}\mathfrak{n}\mathfrak{é}$ s'il y a une suite de coups $[\boxtimes_\xi, \boxtimes]$;
 - la divergence négative \mathfrak{Fid}^- sinon.

Proposition 7.2 – Traduction et normalisation

La traduction par les arbres de Böhm abstraits commute avec la normalisation.

On en déduit la validité des définitions.

Cette traduction permet de montrer un demi-principe des équations génériques.

Proposition 7.3 – Demi-principe des équations génériques

Toute équation générique E vraie sur les arbres de Böhm abstraits l'est aussi en ludique avec pointeurs.

Démonstration : C'est une conséquence du codage : la normalisation commute à la traduction.

Soit E vraie sur les arbres de Böhm abstraits. Soient $\mathcal{D}_{i_1}, \dots, \mathcal{D}_{i_n}$ des desseins avec pointeurs et $\mathfrak{R}_{j_1}, \dots, \mathfrak{R}_{j_m}$ des réseaux avec pointeurs tels que les deux membres de l'équation E sont bien formés. Soient $\tilde{\mathcal{D}}_{i_1}, \dots, \tilde{\mathcal{D}}_{i_n}$ et $\tilde{\mathfrak{R}}_{j_1}, \dots, \tilde{\mathfrak{R}}_{j_m}$ leurs traductions dans les arbres de Böhm abstraits. Les desseins $\tilde{\mathcal{D}}_{i_1}, \dots, \tilde{\mathcal{D}}_{i_n}$ et les réseaux $\tilde{\mathfrak{R}}_{j_1}, \dots, \tilde{\mathfrak{R}}_{j_m}$ sont tels que $E(\tilde{\mathcal{D}}_i, \tilde{\mathfrak{R}}_j)$ est bien formée. Par hypothèse, $E(\tilde{\mathcal{D}}_i, \tilde{\mathfrak{R}}_j)$ est vraie. Donc, par injectivité de la traduction, $E(\mathcal{D}_i, \mathfrak{R}_j)$ est vraie donc E est vérifiée en ludique avec pointeurs. ■

Remarque 7.3 On n'a pas de principe des équations génériques (proposition 3.10 page 96) car les arbres de Böhm abstraits ne sont pas des desseins mais la moitié utile de la proposition se conserve : les équations génériques vraies dans les arbres de Böhm abstraits sont vraies en ludique avec pointeurs.

Comme pour la ludique probabiliste, cette proposition permet de montrer le théorème d'associativité (proposition 7.6 page 193).

7.5 Plongement de la ludique simple

La ludique simple se plonge naturellement en ludique avec pointeurs en mettant des pointeurs aux actions. On remarque que la convention de comptage des actions permet de définir très simplement des fonctions : il suffit de mettre des 0 aux bons endroits.

Définition 7.27 – La fonction Point

La fonction Point de mise des pointeurs est la fonction qui prend un dessein simple sur une base β et renvoie un dessein avec pointeurs de la manière suivante :

$$\begin{array}{lll}
 \boxtimes & \mapsto & \boxtimes \\
 \boxtimes_{\xi} & \mapsto & \begin{array}{l} \boxtimes_{\xi} \\ \boxtimes_{0:\xi} \end{array} \quad \begin{array}{l} \text{si } \xi \in \beta \\ \text{sinon} \end{array} \\
 (+, \xi, I) & \mapsto & \begin{array}{l} (+, \xi, I) \\ (+, 0 : \xi, I) \end{array} \quad \begin{array}{l} \text{si } \xi \in \beta \\ \text{sinon} \end{array} \\
 (-, \xi, I) & \mapsto & (-, \xi, I)
 \end{array}$$

Proposition 7.4 – Compatibilité du plongement

Soit \mathfrak{R} un réseau simple. On a :

$$\llbracket \text{Point}(R) \rrbracket = \text{Point}(\llbracket \mathfrak{R} \rrbracket)$$

7.6 Ordres

Les ordres stable et extensionnel de la ludique simple (section 1.2.5 page 34) sont généralisés naturellement. L'ordre extensionnel n'est pas caractérisé comme en ludique simple (proposition 1.1 page 35) car la ludique avec pointeur n'est pas séparée (section 7.7.1 page suivante).

Définition 7.28 – Ordre stable

L'ordre stable est l'ordre engendré par :

– Cas positif.

$$\mathfrak{Fid} \sqsubseteq \mathfrak{D}$$

– Cas négatif. si $N \subseteq M$ alors

$$((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in N} \sqsubseteq ((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in M}$$

Définition 7.29 – Ordre extensionnel

L'ordre extensionnel est l'ordre engendré par

– Cas positif.

$$\mathfrak{Fid} \preceq \mathfrak{D}^+ \quad \mathfrak{D}^+ \preceq \mathfrak{K} \quad (+, \xi, I). \mathfrak{R} \preceq \mathfrak{K}_\xi$$

– Cas négatif.

$$\mathfrak{Fid}^- \preceq \mathfrak{D}^-$$

et, si $N \subseteq M$ alors

$$((-, \xi, I). \mathfrak{D}_I)_{I \in N} \preceq ((-, \xi, I). \mathfrak{D}_I)_{I \in M}$$

7.7 À propos des théorèmes analytiques

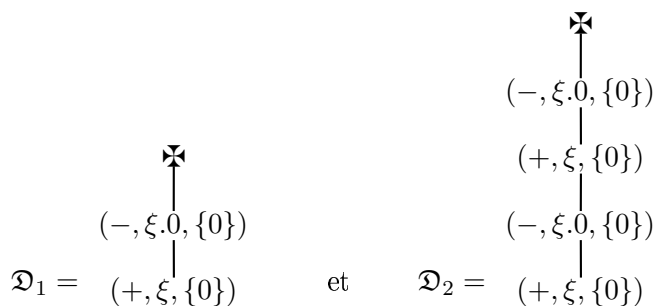
Les théorèmes analytiques ne sont pas tous vérifiés : la séparation est fausse. Cependant, les autres théorèmes se conservent et servent pour les connecteurs ainsi que pour le chapitre 8 page 205.

7.7.1 Non-séparation

Proposition 7.5 – Non-séparation

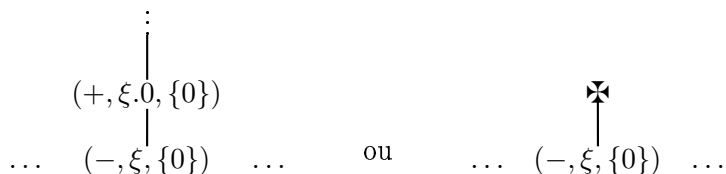
La séparation n'est pas vérifiée en ludique avec pointeurs.

Démonstration : Les desseins \mathfrak{D}_1 et \mathfrak{D}_2 suivants



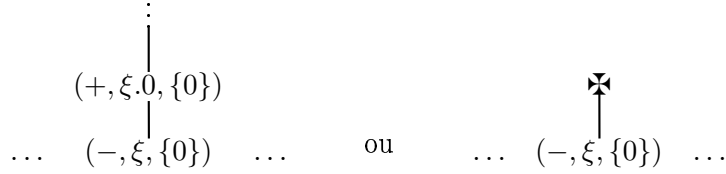
ne sont pas séparables.

En effet, soit \mathfrak{E} un dessin de base $\xi \vdash$. Les desseins \mathfrak{E} et \mathfrak{D}_1 sont orthogonaux si et seulement si \mathfrak{E} est de la forme



Dans les deux cas, \mathfrak{E} est aussi orthogonal à \mathfrak{D}_2 .

Réciproquement, soit \mathfrak{E} un dessin orthogonal à \mathfrak{D}_2 . Il est aussi de la forme



et est donc aussi orthogonal à \mathfrak{D}_1 . ■

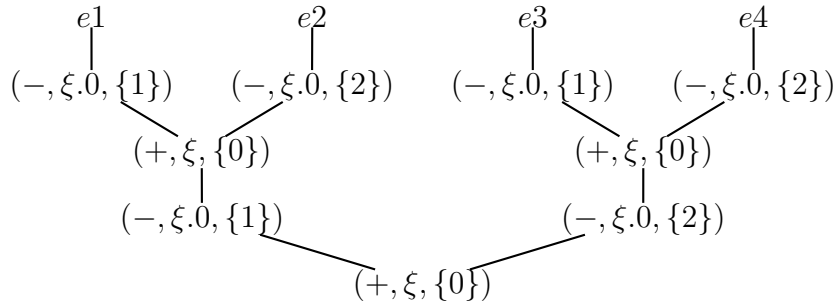
Remarque 7.4 *Le contre-exemple à la séparation est relié au programme*

```

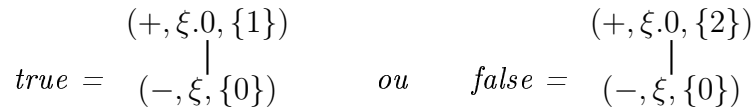
if b then
  if b then e1
  else e2
else
  if b then e3
  else e4
    
```

pour lequel les expressions e2 et e3 ne sont jamais visitées dans une sémantique uniforme (dans laquelle b est toujours true ou toujours false).

Une traduction d'un tel dessein en ludique avec pointeurs serait



Le dessein correspondant à b étant



Les desseins e2 et e3 ne sont pas visités interactivement.

Cette non-séparation pose problème et amène à définir la ludique probabiliste avec pointeurs qu'on appelle *ludique exponentielle* au chapitre 8 page 205.

7.7.2 Associativité

On réutilise la définition 1.35 page 36 des réseaux de réseaux dans le cadre de la ludique avec pointeurs.

Définition 7.30 – Réseau de réseaux

Un **réseau de réseaux** $\{\mathfrak{R}_1, \dots, \mathfrak{R}_n\}$ est une partition d'un réseau en réseaux.

Proposition 7.6 – Associativité

Pour un réseau de réseaux $\{\mathfrak{R}_1, \dots, \mathfrak{R}_n\}$, on a

$$[[\mathfrak{R}_1, \dots, \mathfrak{R}_n]] = [[[\mathfrak{R}_1]], \dots, [\mathfrak{R}_n]]$$

Démonstration : Pour la ludique probabiliste, on a utilisé le principe des équations génériques 3.26 page 110. Ici, on utilise le demi-principe des équations génériques (proposition 7.3 page 189). ■

7.7.3 Stabilité

Définition 7.31 – Inclusion

L'**inclusion** \subseteq entre desseins est synonyme de l'ordre stable.

Remarque 7.5 L'inclusion des chroniques n'est pas l'inclusion des chroniques en temps que suites d'actions car cela impliquerait

$$\begin{array}{ccc}
 \begin{array}{c} \mathfrak{S} \mathfrak{k} \mathfrak{u} \mathfrak{n} \mathfrak{k} \\ | \\ (+, \xi.0.0, \{0\}) \\ | \\ (-, \xi.0, \{0\}) \\ | \\ (+, \xi, \{0\}) \end{array} & & \begin{array}{c} \mathfrak{S} \mathfrak{i} \mathfrak{d}^- \\ | \\ (+, \xi.0.0, \{0\}) \\ | \\ (-, \xi.0, \{0\}) \\ | \\ (+, \xi, \{0\}) \end{array} \\
 = & & \subseteq
 \end{array}$$

ce qui contreviendrait à l'ordre extensionnel.

Définition 7.32 – Compatibilité

Deux desseins sont **compatibles** lorsqu'ils sont inclus dans un même dessein.

Définition 7.33 – Intersection

L'**intersection** $\bigcap_i \mathfrak{D}_i$ d'une famille (\mathfrak{D}_i) de desseins compatibles est le dessein formé des chroniques incluses dans chaque \mathfrak{D}_i .

On étend naturellement cette notion aux réseaux.

Proposition 7.7 – Stabilité

Soit (\mathfrak{R}_i) une famille de réseaux inclus dans un réseau \mathfrak{R} . On a

$$\llbracket \bigcap \mathfrak{R}_i \rrbracket = \bigcap \llbracket \mathfrak{R}_i \rrbracket$$

Démonstration : On passe par les historiques.

1. On vérifie que les historiques de l'intersection des \mathfrak{R}_i est l'intersection des historiques des réseaux \mathfrak{R}_i .
2. On en déduit que les oublis des historiques de l'intersection des \mathfrak{R}_i est l'intersection des oublis des historiques des réseaux \mathfrak{R}_i .
3. On en conclut que $\llbracket \bigcap \mathfrak{R}_i \rrbracket = \bigcap \llbracket \mathfrak{R}_i \rrbracket$. ■

Remarque 7.6 *La preuve de Jean-Yves Girard [Gir01] passe par des “pull-backs” ce qui correspond à ces notions d'historiques.*

7.7.4 Monotonie**Proposition 7.8 – Monotonie**

La normalisation est croissante vis-à-vis de l'ordre extensionnel \preceq .

$$\forall i, \mathfrak{D}_i \preceq \mathfrak{D}'_i \implies \llbracket \mathfrak{D}_1, \dots, \mathfrak{D}_n \rrbracket \preceq \llbracket \mathfrak{D}'_1, \dots, \mathfrak{D}'_n \rrbracket$$

Démonstration : On le vérifie au niveau des historiques. ■

7.8 Comportements

Pour interpréter la logique linéaire, on définit deux notions de comportements : les comportements multiples et les comportements linéaires. Contrairement à un comportement multiple, un comportement linéaire est composé de desseins ne répétant pas les lieux de la base.

Cette nomenclature est nécessaire pour définir simplement les connecteurs positifs. Elle se justifie aussi par la nature différente des connecteurs de la logique linéaire habituels (y compris les exponentielles) et les connecteurs \multimap et \sharp (section 7.9.2 page 200). Ces derniers seront interprétés dans des comportements multiples alors que les connecteurs habituels le sont dans des comportements linéaires.

Les connecteurs habituels de la logique linéaire envoient les comportements linéaires sur les comportements linéaires.

7.8.1 Définitions

Par simplicité, les bases sont a priori supposées unaires.

Définition 7.34 – Éthique

Une **éthique** est un ensemble E non vide de desseins totaux sur une même base.

Définition 7.35 – Orthogonal d'une éthique

L'**orthogonal** d'une éthique est

$$E^\perp = \{\mathcal{D} \mid \forall \mathcal{E} \in E, \mathcal{D} \perp \mathcal{E}\}$$

Définition 7.36 – Comportement multiple

Un **comportement multiple** est une éthique égale à son bi-orthogonal.

Remarque 7.7 L'orthogonal d'une éthique est un comportement multiple.

On remarque (par définition de \preceq) qu'un comportement multiple est clos par l'ordre extensionnel et par l'ordre stable :

Proposition 7.9 – Clôture

- Si $\mathcal{D} \preceq \mathcal{D}'$ et $\mathcal{D} \in G$ alors $\mathcal{D}' \in G$.
- En particulier, si $\mathcal{D} \sqsubseteq \mathcal{D}'$ et $\mathcal{D} \in G$ alors $\mathcal{D}' \in G$.
- Soit (\mathcal{D}_k) une famille de desseins dans un comportement multiple G , si $\bigcup_k \mathcal{D}_k$ est un dessein alors $\bigcap_k \mathcal{D}_k$ est un dessein et appartient à G .

Démonstration : Le dernier point se prouve en considérant un dessein \mathcal{E} de G^\perp et les historiques des réseaux $\mathcal{D}_i, \mathcal{E}$. ■

Définition 7.37 – Ensemble des desseins linéaires sur la base

Un dessein \mathcal{D} sur une base β est **linéaire** si aucun lieu de β n'est utilisé au moins deux fois dans une tranche de \mathcal{D} .

Soit E une éthique sur une base β . On note $\text{Lin}(E)$ l'**ensemble des desseins de E linéaires sur β** .

Définition 7.38 – Bi-orthogonal linéaire

Le **bi-orthogonal linéaire** d'une éthique E est

$$\text{Lin}(E^{\perp\perp})$$

Définition 7.39 – Comportement linéaire

Un **comportement linéaire** est une éthique égale à son bi-orthogonal linéaire.

C'est-à-dire une éthique G telle que

$$G = \text{Lin}(G^{\perp\perp})$$

Remarque 7.8 Les desseins négatifs de base $\xi \vdash$ sont linéaires. En effet, pour pouvoir rejouer une action $(-, \xi, I)$, il faudrait pouvoir utiliser le lieu préfixe de ξ . Les notions de comportement négatif et de comportement négatif linéaire sont donc identiques.

La notion de dessein linéaire n'est pas équivalente à la notion de dessein simple. En effet, un dessein linéaire peut répéter des lieux en dehors de sa base.

La proposition suivante permet de définir l'incarnation indépendamment des questions de linéarité.

Proposition 7.10 – Dessein inclus dans un dessein linéaire

Tout dessein inclus dans un dessein linéaire est linéaire.

Dans le cadre des pointeurs, il y a toujours une notion similaire d'incarnation.

Proposition 7.11 – Incarnation

Pour tout comportement multiple ou linéaire G , et tout dessein \mathcal{D} de G , il existe un unique dessein $\mathcal{D}_0 \subseteq \mathcal{D}$ de G tel que

$$\forall \mathcal{E} \in G, \mathcal{E} \subseteq \mathcal{D} \Rightarrow \mathcal{D}_0 \subseteq \mathcal{E}$$

De plus, ce dessein \mathcal{D}_0 est l'intersection des desseins de G inclus dans \mathcal{D} .

Définition 7.40 – Incarnation

L'**incarnation** d'un dessein \mathcal{D} dans un comportement multiple ou linéaire G est le dessein de la proposition 7.11.

$$|\mathcal{D}|_G = \bigcap \{ \mathcal{E} \in G \mid \mathcal{E} \subseteq \mathcal{D} \}$$

Définition 7.41 – Dessein incarné

Un dessein est **incarné** dans un comportement si il est égal à son incarnation dans ce comportement. On dit aussi qu'un dessein incarné est **matériel**.

Définition 7.42 – Incarnation d'un comportement

L'**incarnation d'un comportement**, notée $|G|$, est l'ensemble de ses desseins incarnés.

Pour définir les connecteurs on donne deux notions d'éthique complète.

Définition 7.43 – Éthique multiple complète

Une éthique est **complète** si $|E^{\perp\perp}| \subseteq E$.

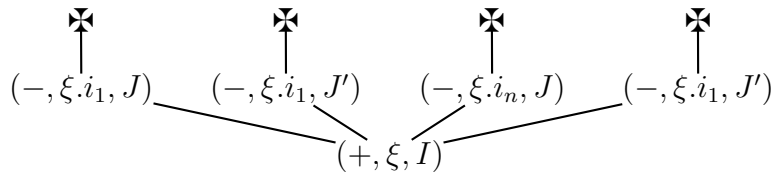
Définition 7.44 – Éthique linéaire complète

Une éthique est une **éthique linéaire complète** si $\text{Lin}(|E^{\perp\perp}|) \subseteq \text{Lin}(E) = E$.

7.8.2 Notions pour la complétude interne

Définition 7.45 – Répertoire d'un comportement

- Le **répertoire** d'un comportement multiple ou linéaire positif G est l'ensemble des ramifications I telles que le dessein suivant appartienne à G



où $\begin{cases} I = \{i_1, \dots, i_n\} \\ J, J' \text{ parcourent } \mathcal{P}_{fin}(\mathbb{N}) \end{cases}$

- Le **répertoire** d'un comportement négatif est l'ensemble des ramifications I

telles que le dessein $\begin{matrix} \text{⊠} \\ \uparrow \\ (-, \xi, I) \end{matrix}$ est inclus dans l'incarnation dans G de

$\begin{matrix} \text{⊠} & & \text{⊠} \\ \uparrow & & \uparrow \\ (-, \xi, J) & \cdots & (-, \xi, J') \end{matrix}$ où $J, J' \text{ parcourent } \mathcal{P}_{fin}(\mathbb{N})$.

- le **répertoire** d'une éthique est le répertoire de son bi-orthogonal.

Proposition 7.12 – Caractérisations du répertoire d'un comportement positif

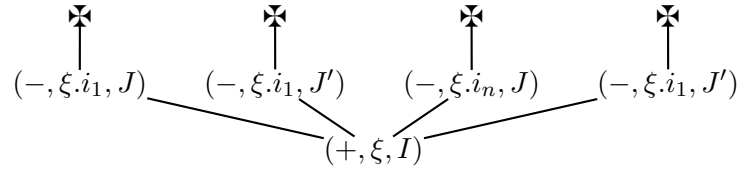
Le répertoire d'un comportement multiple ou linéaire positif G sur la base $\vdash \xi$ est :

1. l'ensemble des ramifications I tel qu'un dessein de G commence par $(+, \xi, I)$;
2. l'ensemble des ramifications I tel qu'un dessein de G a une action $(+, \xi, I)$ visitée par G^\perp .

Démonstration :

1. Par monotonie, le répertoire d'un comportement positif sur la base $\vdash \xi$ contient l'ensemble des ramifications I tel qu'un dessein de G commence par $(+, \xi, I)$.

La réciproque est évidente : si le dessein



appartient à G alors il y a bien un dessin de G qui commence par $(+, \xi, I)$.

2. Ce point découle du point 1. ■

Proposition 7.13 – Répertoire

Tout comportement multiple ou linéaire positif sur une base unaire a le même répertoire que son orthogonal.

Démonstration : La preuve est similaire à la preuve en ludique simple. ■

Définition 7.46 – Comportements disjoints et connexes

- Deux comportements multiples ou linéaires sont **disjoints** si leurs répertoires sont disjoints.
- Une éthique E est **connexe** si son répertoire est un singleton $\{I\}$. L'ensemble I est appelé la **ramification** de E .

Proposition 7.14 – Caractérisation des comportements disjoints

- Deux comportements positifs sont disjoints si et seulement si leur intersection est réduite au démon focalisé.
- Deux comportements négatifs sont disjoints si et seulement si

$$\forall \mathcal{D} \in G \forall \mathcal{E} \in H, |\mathcal{D}|_G \cap |\mathcal{E}|_H = \emptyset.$$

Définition 7.47 – Réservoir

*Un **réservoir** est un ensemble d'entiers.
Le **réservoir** d'un comportement est l'union des ramifications de son répertoire.*

Définition 7.48 – Indépendance

*Deux répertoires R_1 et R_2 sont **indépendants** si*

$$\forall I_1, J_1 \in R_1, I_2, J_2 \in R_2, I_1 \cup I_2 = J_1 \cup J_2 \Rightarrow I_1 = J_1, I_2 = J_2$$

*Deux comportements sont **indépendants** si leurs répertoires sont indépendants.*

Définition 7.49 – Comportements étrangers

*Deux comportements sont **étrangers** si leurs réservoirs sont disjoints.*

7.9 Connecteurs

Les connecteurs additifs et multiplicatifs sont définis de manière similaire à ceux de la ludique simple. Les exponentielles remplacent naturellement les décalages car ces connecteurs sont ici identifiés. La “faute” revient aux décalages : la proposition d’éthique complète linéaire 7.15 fait apparaître un bi-orthogonal.

7.9.1 Décalages

Définition 7.50 – Décalage

– Si \mathcal{D} est un dessein positif de base $\vdash \xi.i$ alors

$$\uparrow\mathcal{D} = \begin{cases} (-, \xi, \{i\}).\mathcal{D} & \text{si } \mathcal{D} \neq \mathfrak{Fid} \\ \mathfrak{Gtunk} & \text{sinon} \end{cases}$$

– Si \mathcal{D} est un dessein négatif de base $\xi.i \vdash$ alors

$$\downarrow\mathcal{D} = (+, \xi, \{i\}).\mathcal{D}$$

– Sur une base unaire, le **décalage** d’un comportement linéaire G est le bi-orthogonal linéaire des décalages de ses desseins. Lorsque G est positif de base $\vdash \xi.i$ (resp. négatif de base $\xi.i \vdash$), on le note $\uparrow G$ (resp. $\downarrow G$).

Proposition 7.15 – Complétude linéaire interne pour les décalages

Soit G un comportement linéaire.

- Si G est positif, alors $\{\uparrow\mathcal{D} \mid \mathcal{D} \in G^{\perp\perp}\}$ est une éthique complète pour $\uparrow G$.
- Si G est négatif, alors $\downarrow G = \{\downarrow\mathcal{D} \mid \mathcal{D} \in G\} \cup \{\mathfrak{X}_\xi\}$.

Démonstration :

- Soit $\mathcal{E} \in \uparrow G$ un dessein négatif.
Soit $\mathcal{E}' \in G^\perp$ et $\mathcal{D} = (+, \xi, \{i\}).\mathcal{E}'$.
On a $\mathcal{E} \perp \mathcal{D}$ car \mathcal{D} est orthogonal à tous les desseins de $\{\uparrow\mathcal{E} \mid \mathcal{E} \in G\}$.
Donc \mathcal{E} s’écrit $\dots (-, \xi, \{i\}).\mathcal{D}' \dots$ avec $\mathcal{D}' \perp \mathcal{E}'$.
En faisant varier \mathcal{E}' , on déduit que $\mathcal{D}' \in G^{\perp\perp}$.
Donc $\{\uparrow\mathcal{D} \mid \mathcal{D} \in G^{\perp\perp}\}$ est une éthique complète pour $\uparrow G$.
- On a $\downarrow G \supseteq \{\downarrow\mathcal{D} \mid \mathcal{D} \in G\} \cup \{\mathfrak{X}_\xi\}$.
Montrons l’autre inclusion. Soit $\mathcal{D} \in \downarrow G$. Supposons $\mathcal{D} \neq \mathfrak{X}_\xi$. Le dessein \mathcal{D} est orthogonal à

$$\begin{array}{c} \mathfrak{X} \\ \uparrow \\ (-, \xi, \{i\}) \end{array}$$

car tous les desseins de $\{\downarrow\mathcal{D} \mid \mathcal{D} \in G\}$ le sont. Donc \mathcal{D} s’écrit $\downarrow\mathcal{E}$.

Par linéarité, le lieu ξ n’apparaît pas dans \mathcal{E} donc \mathcal{E} est orthogonal à G^\perp .

Donc $\mathcal{D} \in \{\downarrow\mathcal{D} \mid \mathcal{D} \in G\}$ ■

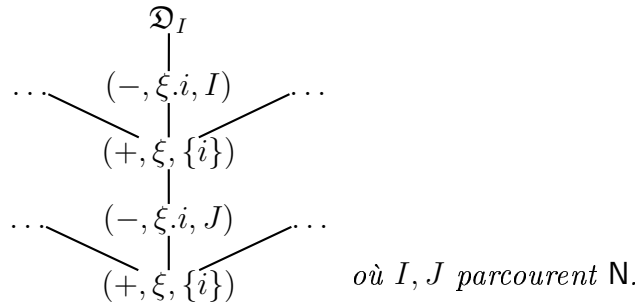
Le bi-orthogonal de la proposition précédente (pour la complétude interne du décalage négatif) montre que la notion interactive naturelle n'est pas celle de décalage : c'est celle d'exponentielle (définition 7.51). Plus exactement, dans le cadre de la ludique avec répétitions, on ne sait pas différencier ces connecteurs. On retrouvera de vrais décalages avec l'uniformité à la section 8.8 page 233.

Remarque 7.9 *Si G est négatif, alors l'égalité*

$$(\downarrow G)^{\perp\perp} = \{\downarrow \mathcal{D} \mid \mathcal{D} \in G\} \cup \{\boxtimes_{\xi}\}$$

est toujours fausse.

En effet, si $((-, \xi.i, I).\mathcal{D}_I)_{I \in \mathbb{N}} \in G$ avec $\mathbb{N} \neq \emptyset$ alors



appartient à $\downarrow G$ mais n'est pas le décalage d'un dessin.

7.9.2 Exponentielles

Ici, les exponentielles sont égales aux décalages. Les exponentielles sont décomposées en deux “connecteurs” : un connecteur pour dupliquer et un connecteur pour changer de polarité. Les changements de polarité sont faits par des décalages, et la partie duplication est prise en compte par les connecteurs \flat et \sharp .

$$!N = \downarrow \sharp N \text{ et } ?P = \uparrow \flat P$$

Dans ce cadre, le \sharp est interprété par l'identité. On le note pour des questions de confort de lecture. Le \flat est le passage d'un comportement linéaire positif à son bi-orthogonal (ce qui explique les guillemets autour de “connecteurs”).

Cette décomposition est naturelle puisqu'elle a déjà été observée au moins par Jean-Yves Girard [Gir01] et Olivier Laurent [Lau04] pour les jeux polarisés.

Définition 7.51 – Exponentielles

- Si G est un comportement linéaire positif, alors $?G$ est le bi-orthogonal linéaire des décalages négatifs des dessins du bi-orthogonal de G .
- Si G est un comportement linéaire négatif, alors $!G$ est le bi-orthogonal linéaire des décalages positifs des dessins de G .

Définition 7.52 – Les connecteurs bémol \flat et dièse \sharp

- Si G est un comportement linéaire positif, on note $\flat G$ son bi-orthogonal.
- Si G est un comportement linéaire négatif, on note $\sharp G$ son bi-orthogonal (égal à G).

Proposition 7.16 – Complétude linéaire interne pour les exponentielles

Soit G un comportement linéaire.

- Si G est positif, alors $\{\uparrow \mathcal{D} \mid \mathcal{D} \in \flat G\}$ est une éthique complète pour $?G$.
- Si G est négatif, alors $!G = \{\downarrow \mathcal{D} \mid \mathcal{D} \in \sharp G\} \cup \{\blackcross_{\xi}\} = \{\downarrow \mathcal{D} \mid \mathcal{D} \in G\} \cup \{\blackcross_{\xi}\}$.

Démonstration : On refait exactement la preuve pour les décalages avec les notations exponentielles.

- Soit $\mathcal{E} \in ?G$ un dessein négatif.
Soit $\mathcal{E}' \in G^{\perp}$ et $\mathcal{D} = (+, \xi, \{i\}).\mathcal{E}'$.
On a $\mathcal{E} \perp \mathcal{D}$ car \mathcal{D} est orthogonal à tous les desseins de $\{\uparrow \mathcal{E} \mid \mathcal{E} \in G\}$.
Donc \mathcal{E} s'écrit $\dots (-, \xi, \{i\}).\mathcal{D}' \dots$ avec $\mathcal{D}' \perp \mathcal{E}'$.
En faisant varier \mathcal{E}' , on déduit que $\mathcal{D}' \in \flat G$.
Donc $\{\uparrow \mathcal{D} \mid \mathcal{D} \in \flat G\}$ est une éthique complète pour $\uparrow G$.
- On a $\downarrow G \supseteq \{\downarrow \mathcal{D} \mid \mathcal{D} \in G\} \cup \{\blackcross_{\xi}\}$.
Montrons l'autre inclusion. Soit $\mathcal{D} \in \downarrow G$. Supposons $\mathcal{D} \neq \blackcross_{\xi}$. Le dessein \mathcal{D} est orthogonal à

\blackcross
 \uparrow
 $(-, \xi, \{i\})$

car tous les desseins de $\{\downarrow \mathcal{D} \mid \mathcal{D} \in G\}$ le sont. Donc \mathcal{D} s'écrit $\downarrow \mathcal{E}$.
Par linéarité, le lieu ξ n'apparaît pas dans \mathcal{E} donc \mathcal{E} est orthogonal à G^{\perp} .
Donc $\mathcal{D} \in \{\downarrow \mathcal{D} \mid \mathcal{D} \in G\}$ ■

Ainsi, l'interprétation que l'on donne ici est :

$$?A = \uparrow A \text{ et } !A = \downarrow A$$

Le défaut n'est pas dans les exponentielles mais dans les décalages.

7.9.3 Additifs

Les additifs sont définis de manière similaire à la ludique simple. Pour le *plus*, comme pour tous les connecteurs binaires positifs, les comportements utilisés sont des comportements linéaires positifs.

Définition 7.53 – Le connecteur \bigcap

Soit (G_k) une famille de comportements multiples (resp. linéaires) sur une même base.
On définit le comportement multiple (resp. linéaire) $\bigcap_k G_k$ par :

$$\bigcap_k G_k = \{\mathcal{D} \mid \forall k, \mathcal{D} \in G_k\}$$

Définition 7.54 – Le connecteur $\bigcup^{\perp\perp}$

Soit (G_k) une famille de comportements multiples (resp. linéaires) sur une même base.
On définit le comportement multiple (resp. linéaire) $\bigcup_k^{\perp\perp} G_k$ par :

$$\bigcup_k^{\perp\perp} G_k = \{\mathfrak{D} \mid \exists k, \mathfrak{D} \in G_k\}^{\perp\perp} \text{ (resp. } \bigcup_k^{\perp\perp} G_k = \text{Lin}(\{\mathfrak{D} \mid \exists k, \mathfrak{D} \in G_k\}^{\perp\perp}))$$

Définition 7.55 – Le connecteur avec $\&$

Le connecteur $\&$ est le connecteur \cap appliqué à des comportements négatifs disjoints.

Le connecteur $\&$ est encore un produit cartésien sur les incarnations :

Proposition 7.17 – Mystère de l'incarnation

Pour tous comportements disjoints G et H sur une même base négative,

$$|G \& H| = |G| \times |H|$$

Définition 7.56 – Le connecteur *plus* \oplus

Le connecteur \oplus est le connecteur $\bigcup^{\perp\perp}$ appliqué à des comportements linéaires positifs disjoints.

Proposition 7.18 – Propriété de la disjonction

Pour tous comportements linéaires positifs disjoints,

$$G \oplus H = G \cup H$$

Remarque 7.10 Le résultat similaire sur les comportements multiples est faux car un dessin de $(G \oplus H)^{\perp\perp}$ peut utiliser dans une même branche des actions $(+, \xi, I)$ et $(+, \xi, J)$ où I appartient au répertoire de G et J appartient au répertoire de H . Cela montre l'utilité de la notion de comportement linéaire.

7.9.4 Multiplicatifs

Définition 7.57 – Le connecteur *tenseur*

Le tenseur \otimes de deux desseins positifs \mathcal{D}_1 et \mathcal{D}_2 est défini, comme pour la ludique simple, par cas sur la première action :

- Si un des desseins est le démon focalisé \mathfrak{X}_ξ , le tenseur est le démon focalisé \mathfrak{X}_ξ .
- Si $\mathcal{D}_1 = (+, \xi, I_1).\mathfrak{R}_1$ et $\mathcal{D}_2 = (+, \xi, I_2).\mathfrak{R}_2$. Il y a deux cas :
 - Si $I_1 \cap I_2 = \emptyset$ alors $\mathcal{D}_1 \otimes \mathcal{D}_2 = (+, \xi, I_1 \cup I_2).\mathfrak{R}_1\mathfrak{R}_2$.
 - Si $I_1 \cap I_2 \neq \emptyset$ alors il y a quatre cas suivant la définition du tenseur :
 - Tenseur commutatif minimal : $\mathcal{D}_1 \otimes \mathcal{D}_2 = \mathfrak{F}id$;
 - Tenseur commutatif maximal : $\mathcal{D}_1 \otimes \mathcal{D}_2 = \mathfrak{X}_\xi$;
 - Tenseur non-commutatif gauche : $\mathcal{D}_1 \otimes \mathcal{D}_2 = (+, \xi, I_1 \cup I_2).\mathfrak{R}_1\mathfrak{R}'_2$ où \mathfrak{R}'_2 est égal à \mathfrak{R}_2 privé des desseins commençant par des actions sur des lieux dans $\{\xi.i \mid i \in I_1\}$;
 - Tenseur non-commutatif droite : $\mathcal{D}_1 \otimes \mathcal{D}_2 = (+, \xi, I_1 \cup I_2).\mathfrak{R}_2\mathfrak{R}'_1$ où \mathfrak{R}'_1 est égal à \mathfrak{R}_1 privé des desseins commençant par des actions sur des lieux dans $\{\xi.i \mid i \in I_2\}$.

Définition 7.58 – Le connecteur \bowtie

Le nœud papillon \bowtie est le dual du tenseur commutatif maximal.

Proposition 7.19 – Existence d'une projection

Soit $\mathbb{M} \subseteq \mathbb{N}$ un réservoir. Tout dessin positif $\mathcal{D} = (+, \xi, I).\mathfrak{R}$ est égal à $\mathcal{D}_{\mathbb{M}} \otimes \mathcal{D}_{\mathbb{N} \setminus \mathbb{M}}$ où $\mathcal{D}_{\mathbb{M}}$ commence par $(+, \xi, I \cap \mathbb{M})$ et $\mathcal{D}_{\mathbb{N} \setminus \mathbb{M}}$ commence par $(+, \xi, I \setminus \mathbb{M})$.

Le dessin $\mathcal{D}_{\mathbb{M}}$ est le **projeté** de \mathcal{D} sur \mathbb{M} .

On étend cette définition en posant $\mathfrak{X}_{\xi_{\mathbb{M}}} = \mathfrak{X}_\xi$.

Démonstration : L'idée de la construction de $\mathcal{D}_{\mathbb{M}}$ est simplement d'écrire

$$\mathcal{D} = (+, \xi, I).(\mathcal{D}_{\xi.i_1}, \dots, \mathcal{D}_{\xi.i_n})$$

et de ne conserver que les $\mathcal{D}_{\xi.i_k}$ avec $i_k \in \mathbb{M}$. ■

Proposition 7.20 – Projection

Soit E une éthique connexe et \mathbb{M} un réservoir. On a $(E^{\perp\perp})_{\mathbb{M}} = (E_{\mathbb{M}})^{\perp\perp}$.

Comme pour la ludique simple, deux comportements étrangers sont indépendants. De plus, dans ce cas, les quatre *tenseurs* sont égaux. Le résultat est noté $G_1 \otimes G_2$.

Proposition 7.21 – Complétude interne pour le *tenseur*

Si G_1 et G_2 sont positifs, linéaires et indépendants alors $\{\mathcal{D}_1 \otimes \mathcal{D}_2 \mid \mathcal{D}_1 \in G_1, \mathcal{D}_2 \in G_2\}$ est une éthique linéaire complète pour $G_1 \otimes G_2$.

Démonstration : Comme pour la ludique simple ou la ludique probabiliste, on utilise la proposition d'existence d'une projection (proposition 7.19). ■

Chapitre 8

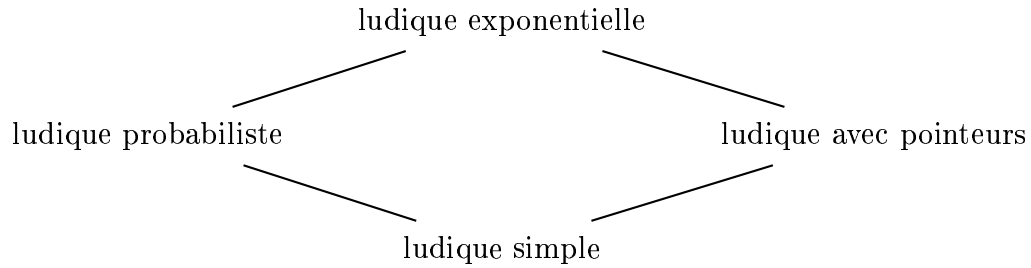
Ludique exponentielle

Sommaire

8.1	Intuitions	206
8.2	Desseins et réseaux	207
8.2.1	Desseins	208
8.2.2	Réseaux	209
8.3	Normalisation	210
8.3.1	Traduction en ludique avec pointeurs	210
8.3.2	Plongement de la ludique avec pointeurs	214
8.4	Ordres	215
8.5	Théorèmes analytiques	216
8.5.1	Associativité	216
8.5.2	Séparation	217
8.5.3	Stabilité	225
8.5.4	Monotonie	226
8.6	Comportements	227
8.6.1	Définitions	227
8.6.2	Notions pour la complétude interne	229
8.7	Connecteurs	230
8.7.1	Décalages	230
8.7.2	Exponentielles	231
8.7.3	Additifs	232
8.7.4	Multiplicatifs	232
8.8	Uniformité	233
8.8.1	Bi-comportements	233
8.8.2	Connecteurs	235
8.8.3	Décalages	235
8.8.4	Exponentielles	236
8.8.5	Additifs	237
8.8.6	Multiplicatifs	237
8.9	Un mot sur l'interprétation	239

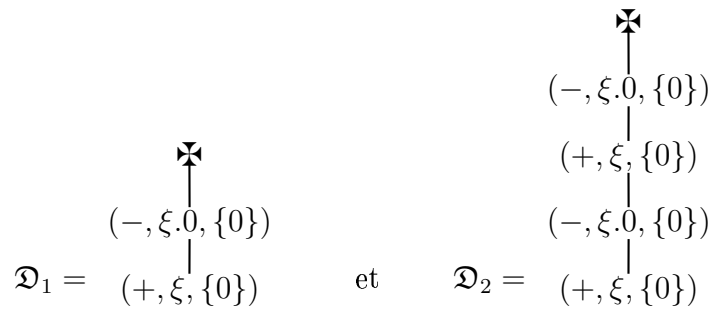
Ce chapitre introduit la *ludique exponentielle*, une proposition de modèle ludique de la logique linéaire.

La non-séparation de la ludique avec pointeurs (section 7.7.1 page 191) impose d'utiliser des nouveaux desseins pour séparer les desseins avec pointeurs : ce sont les desseins probabilistes avec pointeurs. L'idée est de construire une sorte de "losange de ludiques" et de définir la ludique exponentielle comme une sorte de plus petit majorant de la ludique probabiliste et de la ludique avec pointeurs.



8.1 Intuitions

La ludique avec pointeurs n'est pas séparée (proposition 7.5 page 191). La preuve de non-séparation utilise les deux desseins suivants :



qui ont le même orthogonal.

Supposons que l'on sache augmenter la ludique avec pointeurs avec des coefficients comme cela a été fait pour la ludique simple avec la ludique probabiliste (partie II page 65).

Posons

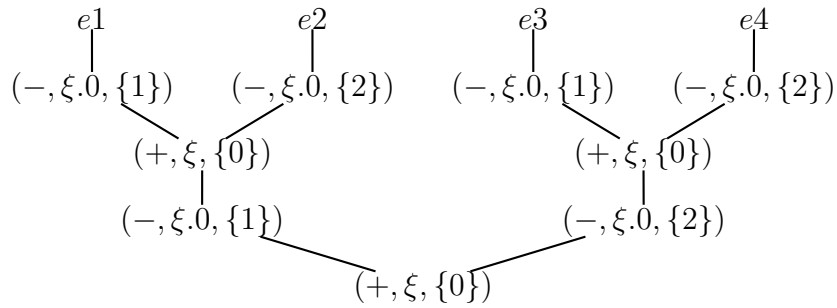
$$\mathfrak{E} = \begin{array}{c} (+, \xi.0, \{0\}) \\ | \\ 0,5 \\ | \\ (-, \xi, \{0\}) \end{array}$$

On a naturellement

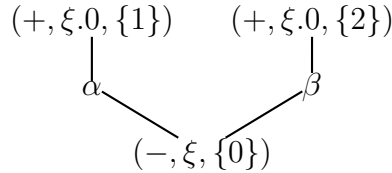
$$[[\mathfrak{D}_1, \mathfrak{E}]] = \begin{array}{c} \blackcross \\ | \\ 0.5 \end{array} \quad \text{et} \quad [[\mathfrak{D}_2, \mathfrak{E}]] = \begin{array}{c} \blackcross \\ | \\ 0.5 \\ | \\ 0.5 \end{array}$$

ce qui sépare \mathfrak{D}_1 et \mathfrak{D}_2 .

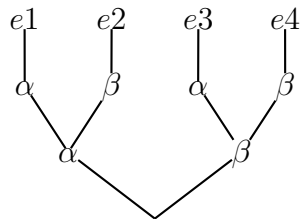
De même, face au dessin



on peut utiliser un booléen probabiliste $\mathbf{b} =$



La normalisation donne alors



ce qui permet d'observer \mathbf{e}_2 et \mathbf{e}_3 .

Cela justifie la définition des coefficients comme des arbres et non des réels.

8.2 Dessins et réseaux

On unifie les définitions de la ludique probabiliste et de la ludique avec pointeurs.

8.2.1 Dessesins

Définition 8.1 – Arbre exponentiel

Un **arbre exponentiel** \mathfrak{D} est un arbre fini ou infini engendré par la grammaire suivante :

arbre	\mathfrak{D}	$:=$	$\mathfrak{D}^+ \mid \mathfrak{D}_\xi^-$
arbre positif	\mathfrak{D}^+	$:=$	$\mathbf{0} \mid \mathfrak{D}_t^+$
arbre positif non vide	\mathfrak{D}_s^+	$:=$	$c.(\mathfrak{D}_s^+, \dots, \mathfrak{D}_s^+)$
arbre positif semi-simple	\mathfrak{D}_s^+	$:=$	$\mathfrak{X} \mid \mathfrak{X}_{\pi_\xi} \mid (+, \pi_\xi, \{i_1, \dots, i_n\}).(\mathfrak{D}_{\xi.i_1}^-, \dots, \mathfrak{D}_{\xi.i_n}^-)$
arbre négatif à la base	\mathfrak{D}_ξ^-	$:=$	$c.(\mathfrak{D}_{s_\xi}^-, \dots, \mathfrak{D}_{s_\xi}^-)$
arbre négatif semi-simple	$\mathfrak{D}_{s_\xi}^-$	$:=$	$((-, \xi, I).\mathfrak{D}_t^+)_{I \in \mathbb{N}}$
pointeur	π_ξ	$:=$	$\xi \mid n : \xi$

où

- \mathbb{N} est un ensemble de sous-parties finies de \mathbb{N}
- $c.(\mathfrak{D}_s^+, \dots, \mathfrak{D}_s^+)$ représente un arbre composé du coefficient c différent de $\mathbf{0}$ suivi par un dessin \mathfrak{D}_s^+ pour chaque feuille vide de c .
- $c.(\mathfrak{D}_{s_\xi}^-, \dots, \mathfrak{D}_{s_\xi}^-)$ représente un arbre composé du coefficient c (possiblement égal à $\mathbf{0}$) suivi par un dessin $\mathfrak{D}_{s_\xi}^-$ pour chaque feuille vide de c .

Un arbre simple est **positif** s'il est de la forme \mathfrak{D}^+ et **négatif**, s'il est de la forme \mathfrak{D}_ξ^- .

Remarque 8.1 On emploie dorénavant le terme *semi-simple* pour des arbres exponentiels dont le coefficient à la base est $\mathbf{1}$.

On utilise aussi la notion de dessin immédiatement total.

Définition 8.2 – Dessin immédiatement total

Un dessin est **immédiatement total** si le poids de son coefficient à la racine est $\mathbf{1}$.

Définition 8.3 – Combinaison linéaire

Soit c un coefficient et α_i les occurrences des feuilles vides de c . Le dessin dont le coefficient à la base est c et les feuilles vides sont associées aux desseins \mathfrak{D}_{α_i} est la **combinaison linéaire** de coefficient c des desseins $(\mathfrak{D}_{\alpha_i})$.

On note les combinaisons linéaires $c.(\mathfrak{D}_{\alpha_i})$.

La notion suivante est la notion similaire au projection simple dans le cadre de la ludique exponentielle.

Définition 8.4 – Projection sans coefficients

Une **projection sans-coefficients** d'un arbre exponentiel est un arbre avec pointeurs obtenu par remplacement de ses coefficients c (et de leurs continuations) par :

- si c est de poids nul, la divergence de la polarité donnée;
- sinon (c a des feuilles vides),
 - la continuation d'une feuille vide de c ;
 - ou éventuellement, la divergence de la polarité donnée si c est de poids différent de 1.

Définition 8.5 – Dessin exponentiel

Un **dessein** sur une base β est un arbre exponentiel sur β tel que ses projections simples sont des desseins exponentiels sur la base β .

On ne détaille pas la définition de la justification qui est définie naturellement.

Définition 8.6 – Ordre d'inclusion

L'**inclusion** entre arbres exponentiels est l'ordre engendré par :

- Cas positif.

$$\mathbf{0} \subseteq \mathfrak{D}$$

- Cas négatif. si $N \subseteq M$ alors

$$((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in N} \subseteq ((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in M}$$

On reprend la définition des c -tranches de la ludique probabiliste pour définir celle de c -chronique.

Définition 8.7 – c -tranche

Une **c -tranche** est un dessein tel que :

- chaque coefficient positif a au plus une feuille vide (i.e. est suivi d'au plus une action positive immédiate);
- et chaque coefficient négatif est suivi par au plus une action négative immédiate.

Définition 8.8 – c -tranche d'un dessein

Une c -tranche \mathfrak{S} est une **c -tranche d'un dessein** \mathfrak{D} si \mathfrak{S} est incluse dans \mathfrak{D} .

Définition 8.9 – c -chronique

Une **c -chronique** est une c -tranche telle que

- chaque action positive $(+, \pi, \{i_1, \dots, i_n\})$ avec $n \geq 1$ est suivie par au plus un coefficient suivi par au moins une action négative immédiate.
- chaque action négative est suivie par un coefficient différent de $\mathbf{1}$ ou par au moins une action positive.

8.2.2 Réseaux

On réutilise la même notion de réseaux...

Définition 8.10 – Pré-réseau

Un **pré-réseau** est un ensemble fini de desseins sur des bases β_i tel qu'un lieu apparaît au plus dans deux bases : une fois positivement et une fois négativement.

Définition 8.11 – Coupure

Une **coupure** dans un pré-réseau \mathfrak{R} est la donnée de deux desseins de \mathfrak{R} dont les bases ont un lieu en commun.

Définition 8.12 – Réseau

Un **réseau** est un pré-réseau dans lequel la relation de coupure est connexe et acyclique.

Définition 8.13 – Base de réseau

La **base** d'un réseau \mathfrak{R} est la base formée de l'ensemble des lieux (avec leur polarité) des bases des desseins de \mathfrak{R} qui n'apparaissent qu'une fois.

Définition 8.14 – Réseau clos

Un réseau est **clos** lorsque sa base est vide.

Définition 8.15 – Actions et lieux visibles

Soit \mathfrak{R} un réseau.

Un lieu est **visible** dans \mathfrak{R} si c'est un sous-lieu d'un lieu de la base de \mathfrak{R} .

Une action dans un dessin de \mathfrak{R} est **visible** si c'est le démon, un démon focalisé ou une action propre dont le lieu visible.

8.3 Normalisation

Comme pour la normalisation en ludique probabiliste (section 3.3.1 page 82), la méthode pour définir la normalisation et accéder à ses propriétés est d'utiliser une traduction dans la version sans coefficients : en ludique avec pointeurs.

Comme pour la normalisation en ludique probabiliste, la normalisation est définie d'abord pour les desseins positivement probabilistes.

Définition 8.16 – Dessin positivement probabiliste

Un dessin est **positivement probabiliste** si ses coefficients négatifs sont égaux à 1.

8.3.1 Traduction en ludique avec pointeurs

Comme en ludique probabiliste, on utilise une traduction en deux temps : traduction des desseins positivement probabilistes en ludique avec pointeurs puis traduction des des-

seins exponentiels en desseins exponentiels positivement probabilistes. La seule différence est qu'il faut mettre des pointeurs $0 : \xi_c$ et non simplement des lieux ξ_c .

On utilise le même dessein pour coder les scalaires.

Définition 8.17 – Le dessein $\mathfrak{D}_s(\xi_c)$

Soit $s \in]0; 1[$. Le **codage** du scalaire s est le dessein $\mathfrak{D}_s(\xi_c)$:

$$\begin{array}{c}
 \vdots \\
 | \\
 (+, 0 : \xi.s_1.s_2.s_3, \{s_4\}) \\
 | \\
 (-, \xi_c.s_1.s_2, \{s_3\}) \\
 | \\
 (+, 0 : \xi_c.s_1, \{s_2\}) \\
 | \\
 (-, \xi_c, \{s_1\})
 \end{array}$$

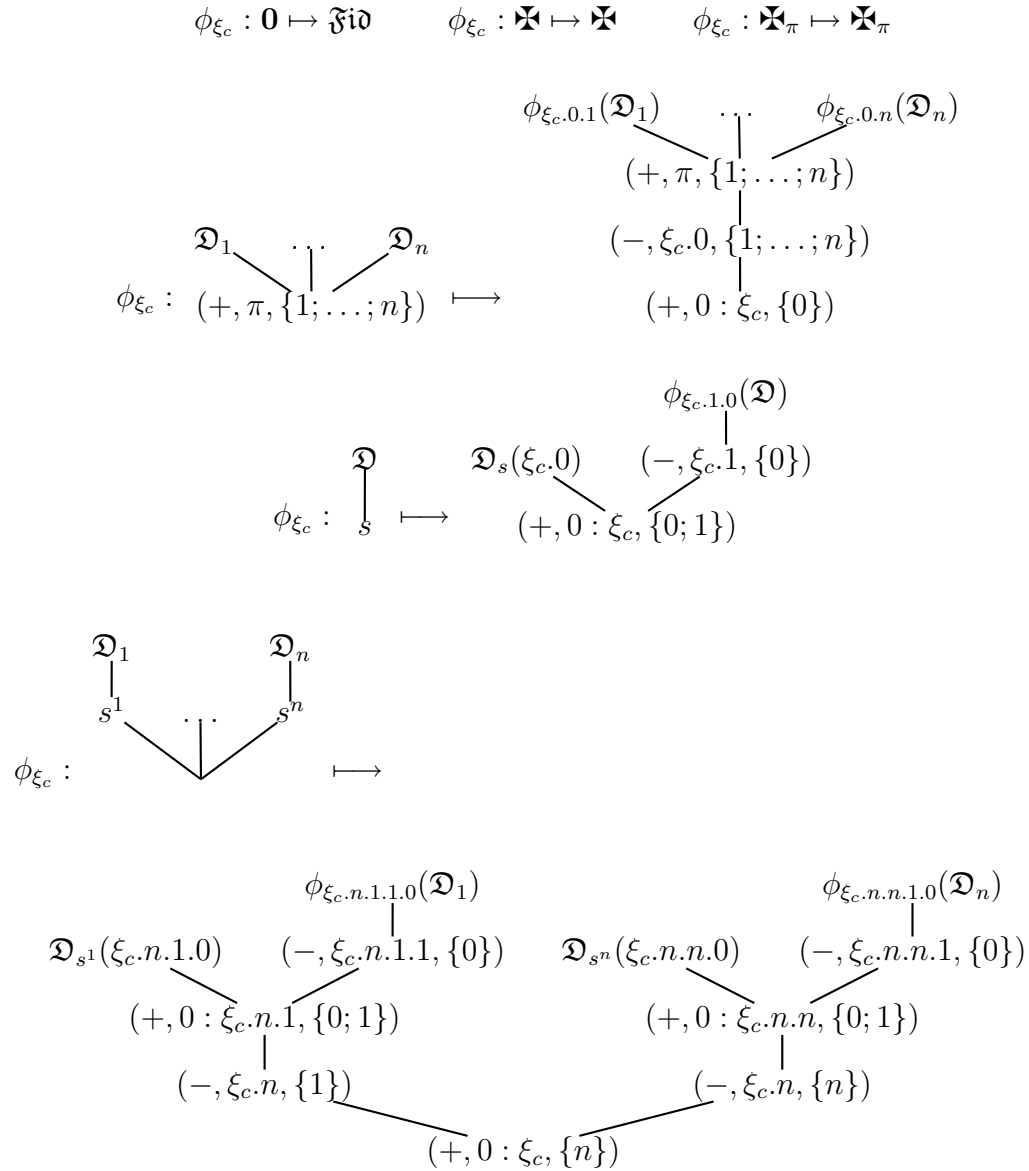
où $s = 0.s_1s_2\dots$

La traduction des desseins n'est pas plus compliquée que pour la ludique probabiliste. En particulier, les pointeurs $n : \xi.i$ n'ont pas à être renommés car l'entier n compte les actions négatives de la forme $(-, \xi, J)$ entre $(+, x : \xi.i, I)$ et son justifiant et non toutes les actions négatives.

Dans les deux définitions suivantes, on utilise l'abus de notation consistant à noter $(+, 0 : \xi, I)$ une action $(+, \xi, I)$ si ξ est un lieu de la base.

Définition 8.18 – Traduction vers la ludique avec pointeurs

Étant donné un lieu ξ_c , un dessein exponentiel positif (ou négatif semi-simple) sur une base β disjointe de ξ_c est **traduit** par la fonction récursive ϕ_{ξ_c} de la manière suivante :



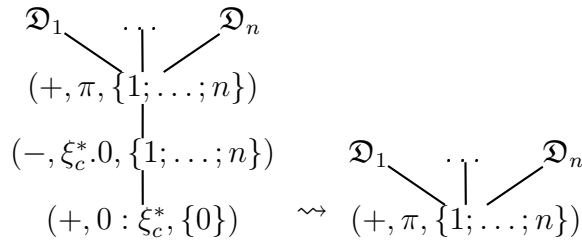
Remarque 8.2 Puisque l'on peut dorénavant répéter les lieux, on pourrait utiliser plusieurs fois ξ_c sans focaliser sur des sous-lieux de longueur arbitraire de ξ_c . Cela n'apporte pas grand chose et cela enlèverait la propriété d'injection immédiate de la ludique probabiliste en ludique exponentielle.

Définition 8.19 – Traduction inverse d'un dessin

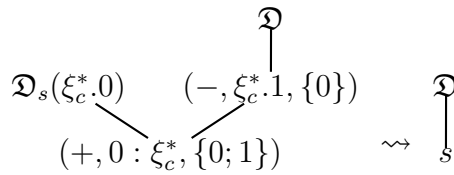
La **traduction inverse** d'un dessin exponentiel traduit est l'unique forme normale obtenue par réécriture (infinie) suivant les règles confluentes suivantes (sur les arbres exponentiels) :

On note ξ_c^* un sous-lieu de ξ_c quelconque.

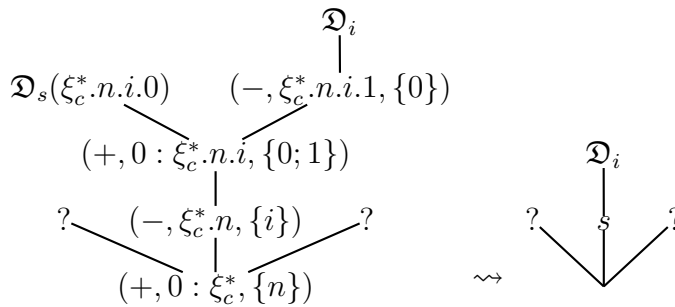
$$\mathfrak{Fid} \rightsquigarrow \mathbf{0}$$



(et le cas sans continuation)
 $(+, 0 : \xi_c^*, \{0\}) \rightsquigarrow \mathbf{0}$



La traduction inverse d'un nœud commençant par $(+, 0 : \xi_c^*, \{n\})$ se fait sur toutes les branches en même temps de la manière suivante (les ? indiquent qu'il faut faire la même opération sur les autres branches) :



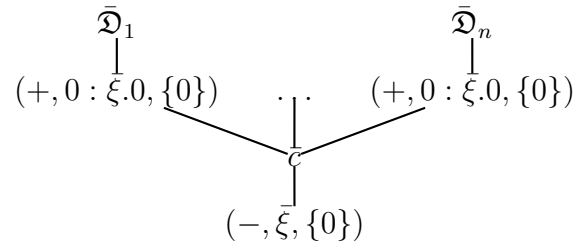
Définition 8.20 – Normalisation en ludique exponentielle

La **normalisation** $[[\mathfrak{R}]]$ d'un réseau exponentiel \mathfrak{R} de desseins positivement probabilistes s'effectue ainsi :

1. Associer un réseau avec pointeurs \mathfrak{R}_t par la traduction.
2. Normaliser le réseau avec pointeurs \mathfrak{R}_t en un dessein avec pointeurs \mathfrak{D}_t .
3. Remettre les scalaires (définition 8.19 page précédente) sur \mathfrak{D}_t . On obtient un dessein $\mathfrak{D} = [[\mathfrak{R}]]$.

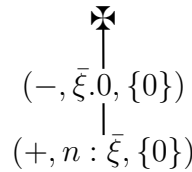
La traduction d'un dessein exponentiel en un dessein exponentiel positivement probabiliste s'effectue comme en ludique probabiliste. Il faut simplement rajouter ou changer les cas suivants :

- un dessein négatif $c.(\mathfrak{D}_1, \dots, \mathfrak{D}_n)$ sur un lieu ξ est traduit en

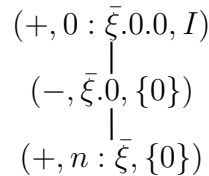


où \bar{c} est égal à c dans lequel les feuilles nulles ont été remplacées par des actions $(+, 0 : \bar{\xi}.0, \emptyset)$. Cette étape est expliquée en ludique probabiliste à la remarque 3.8 page 94.

- une action $\bar{\mathfrak{X}}_{n;\xi}$ est traduite en



- une action $(+, n : \xi, I)$ est traduite en



Comme en ludique probabiliste, la proposition suivante est vérifiée.

Proposition 8.1 – Monotonie de la traduction inverse

La traduction inverse est monotone.

8.3.2 Plongement de la ludique avec pointeurs

Définition 8.21 – Plongement canonique des desseins avec pointeurs

Le **plongement canonique** des desseins avec pointeurs dans les desseins exponentiels consiste à traduire les actions positives σ en le coefficient **1** suivi de σ et un nœud négatif par le coefficient **1** suivi de ce nœud.

On obtient naturellement (par traduction) la proposition suivante.

Proposition 8.2 – Plongement des desseins simples

Le plongement canonique des desseins avec pointeurs commute avec la normalisation.

Rappelons que les équations génériques (définition 3.33 page 95) sont définies par la grammaire :

$$\begin{aligned} E &::= \mathfrak{R} = \mathfrak{R} \\ \mathfrak{R} &::= \mathfrak{X} \mid \mathfrak{X}_\xi \mid \mathfrak{Fid} \mid \mathfrak{D}_i \mid \mathfrak{D}_i, \mathfrak{R} \mid \llbracket \mathfrak{R} \rrbracket \mid \mathfrak{R}_j \mid \mathfrak{R}_j, \mathfrak{R} \end{aligned}$$

où les $\mathfrak{D}_i, \mathfrak{R}_i$ sont des terminaux.

Le principe des équations génériques (proposition 3.10 page 96) se généralise à la ludique exponentielle (on fait la *même* preuve).

Proposition 8.3 – Principe des équations génériques

Toute équation générique E est vraie sur les desseins avec pointeurs si et seulement si elle est vraie sur les desseins exponentiels.

Démonstration : C'est une conséquence de la normalisation par codage : par définition, la normalisation commute à la traduction.

- ⇐ Soit E vraie sur les desseins exponentiels. Comme les desseins avec pointeurs se plongent dans les desseins exponentiels, E est vraie sur les desseins avec pointeurs.
- ⇒ Soit E vraie sur les desseins avec pointeurs. Soient $\mathfrak{D}_{i_1}, \dots, \mathfrak{D}_{i_n}$ des desseins exponentiels et $\mathfrak{R}_{j_1}, \dots, \mathfrak{R}_{j_m}$ des réseaux exponentiels tels que les deux membres de l'équation E sont bien formés. Soient $\tilde{\mathfrak{D}}_{i_1}, \dots, \tilde{\mathfrak{D}}_{i_n}$ et $\tilde{\mathfrak{R}}_{j_1}, \dots, \tilde{\mathfrak{R}}_{j_m}$ leurs traductions dans les desseins avec pointeurs. Les desseins $\tilde{\mathfrak{D}}_{i_1}, \dots, \tilde{\mathfrak{D}}_{i_n}$ et les réseaux $\tilde{\mathfrak{R}}_{j_1}, \dots, \tilde{\mathfrak{R}}_{j_m}$ sont tels que $E(\tilde{\mathfrak{D}}_i, \tilde{\mathfrak{R}}_j)$ est bien formée. Par hypothèse, $E(\tilde{\mathfrak{D}}_i, \tilde{\mathfrak{R}}_j)$ est vraie. Donc, par injectivité de la traduction, $E(\mathfrak{D}_i, \mathfrak{R}_j)$ est vraie donc E est vérifiée sur les desseins exponentiels. ■

8.4 Ordres

Les ordres stable et extensionnel sont définis comme en ludique probabiliste.

Définition 8.22 – Ordre stable

L'ordre stable est l'ordre engendré par :

– Cas positif.

$$\mathbf{0} \sqsubseteq \mathfrak{D}$$

– Cas négatif. si $N \subseteq M$ alors

$$((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in N} \sqsubseteq ((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in M}$$

Définition 8.23 – Ordre extensionnel

L'ordre extensionnel est l'ordre engendré par

– Cas positif.

$$\mathbf{0} \preceq \mathfrak{D}^+ \quad \mathfrak{D}^+ \preceq \mathfrak{X} \quad (+, \xi, I) \cdot \mathfrak{R} \preceq \mathfrak{X}_\xi$$

– Cas négatif.

$$\mathbf{0} \preceq \mathfrak{D}^-$$

et, si $N \subseteq M$ alors

$$((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in N} \preceq ((- , \xi, I) \cdot \mathfrak{D}_I)_{I \in M}$$

8.5 Théorèmes analytiques

Cette section est consacrée à la preuve des quatre théorèmes analytiques en ludique exponentielle.

8.5.1 Associativité

On réutilise une dernière fois la définition 1.35 page 36 des réseaux de réseaux.

Définition 8.24 – Réseau de réseaux

Un réseau de réseaux $\{\mathfrak{R}_1, \dots, \mathfrak{R}_n\}$ est une partition d'un réseau en réseaux.

Proposition 8.4 – Associativité

Pour un réseau de réseaux $\{\mathfrak{R}_1, \dots, \mathfrak{R}_n\}$, on a

$$[[\mathfrak{R}_1, \dots, \mathfrak{R}_n]] = [[[\mathfrak{R}_1]], \dots, [\mathfrak{R}_n]]]$$

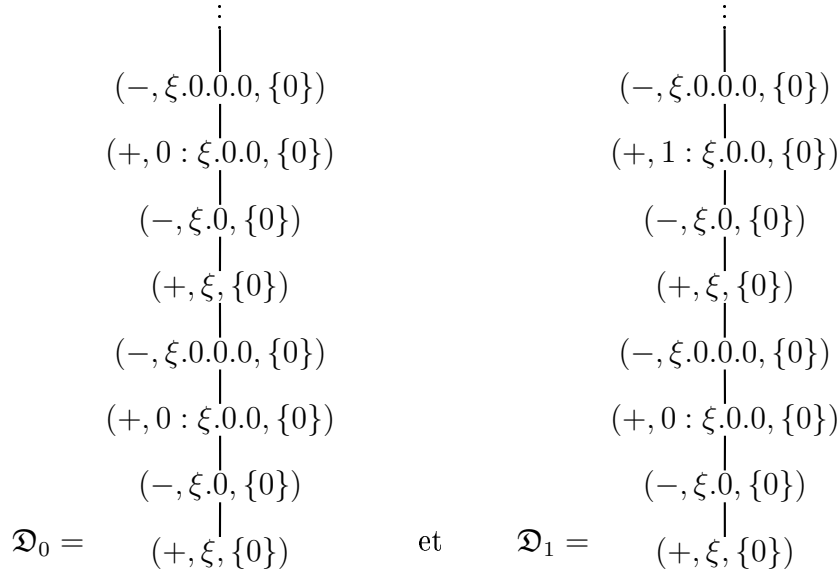
Démonstration : On utilise le principe des équations génériques (proposition 8.3 page précédente). ■

8.5.2 Séparation

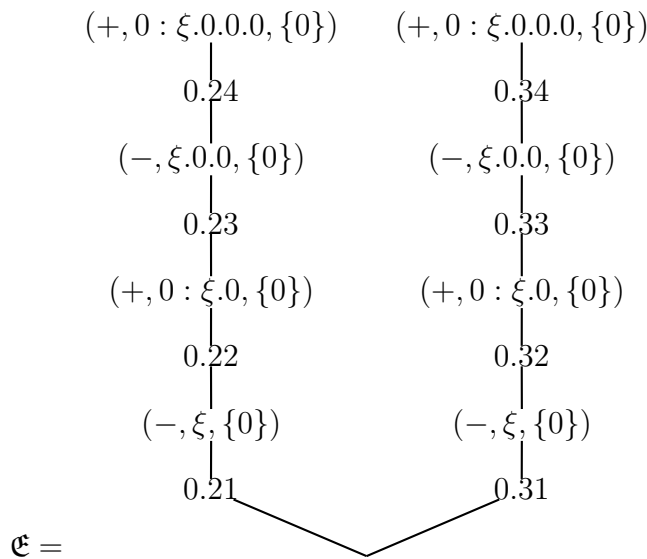
On procède comme en ludique probabiliste : on définit des desseins universels (de manière similaire) puis on montre la séparation.

On commence par un exemple pour montrer que les desseins universels sont plus compliqués et qu'il ne suffit pas de considérer un dessin contenant tous les desseins avec pointeurs (en un certain sens) mais qu'il faut en plus les contenir plusieurs fois à cause des répétitions...

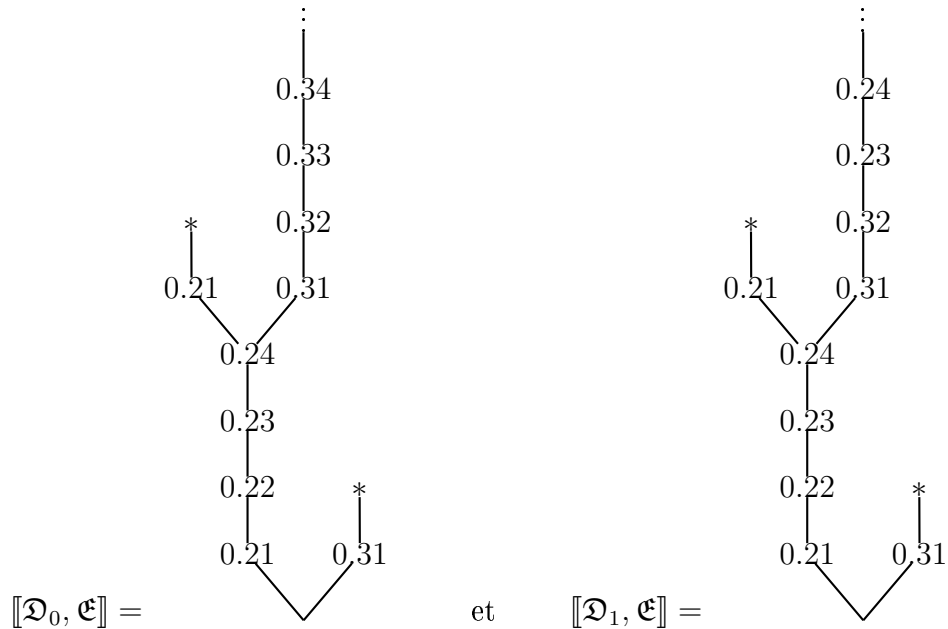
On considère deux desseins



On veut séparer \mathfrak{D}_0 et \mathfrak{D}_1 sans utiliser la partie non visible des desseins (représentée par des \vdots). Pour cela, on peut prendre



Les normalisations sont de la forme



Donc \mathcal{E} sépare \mathcal{D}_0 et \mathcal{D}_1 .

Dans les normalisations de $\mathcal{D}_0, \mathcal{E}$ et $\mathcal{D}_1, \mathcal{E}$, les coefficients notés $*$ représentent des coefficients inutiles pour la séparation. Le phénomène est général (et légèrement simplifié dans cet exemple) : pour séparer \mathcal{D}_0 et \mathcal{D}_1 , il suffit de présenter un dessin de base $\xi \vdash$

- qui branche suffisamment sur les lieux refocalisés (ici on ne l'utilise que pour ξ)
- et d'observer dans le coefficient résultat une branche qui ne répète pas de copies (c'est pour cela qu'on ne regarde pas le coefficient $*$ au dessus du scalaire 0.21).

Cette idée de regarder une branche dans le résultat de la normalisation dans laquelle chaque copie est différenciée par son coefficient préfixe est fondamentale pour la séparation. Elle explique aussi la nécessité de focaliser une infinité de fois à chaque niveau sur chaque lieu dans les desseins universels (définition 8.28 page 222).

Définition 8.25 – Ensemble des scalaires d'un dessin

L'ensemble des scalaires $\text{Scal}(\mathcal{D})$ d'un dessin \mathcal{D} est l'ensemble composé des scalaires apparaissant dans les coefficients de \mathcal{D} .

L'ensemble des scalaires généralisés $\text{Scal}^(\mathcal{D})$ d'un dessin \mathcal{D} est l'ensemble composé des scalaires apparaissant dans les coefficients de \mathcal{D} et des puissances négatives de deux.*

$$\text{Scal}^*(\mathcal{D}) = \text{Scal}(\mathcal{D}) \cup \{2^{-n} \mid n \in \mathbb{N}\}$$

Dans les desseins universels, il faut aussi prendre en compte les répétitions. Pour engendrer les desseins universels en prenant toutes les répétitions en compte, il est alors utile de définir la notion de base interne.

Ainsi, après une pré-chronique

$$\begin{array}{c}
 (-, \xi.0, \{1; 2; 3\}) \\
 | \\
 (+, \xi, \{0\}) \\
 | \\
 (-, \xi.0, \{1; 2\}) \\
 | \\
 (+, \xi, \{0\})
 \end{array}$$

les actions peuvent être de la forme \boxtimes , \boxtimes_ξ , $\boxtimes_{0:\xi.0.1}$, $\boxtimes_{0:\xi.0.2}$, $\boxtimes_{0:\xi.0.3}$, $\boxtimes_{1:\xi.0.1}$, $\boxtimes_{1:\xi.0.2}$, $(+, \xi, I)$, $(+, 0 : \xi.0.1, I)$, $(+, 0 : \xi.0.2, I)$, $(+, 0 : \xi.0.3, I)$, $(+, 1 : \xi.0.1, I)$ ou enfin de la forme $(+, 1 : \xi.0.2, I)$.

La notion de base interne décrit ces ensembles.

Définition 8.26 – Base interne

Une **base interne** est un ensemble de lieux disjoints munis d'une polarité avec au plus un lieu négatif et d'une multiplicité $n \geq 1$ pour certains des lieux positifs.

Une base interne est notée

$$\begin{array}{c}
 \vdash \xi_1, \dots, \xi_k; \xi'_1 \mapsto m_1, \dots, \xi'_n \mapsto m_n \\
 \text{ou} \\
 \xi \vdash \xi_1, \dots, \xi_k; \xi'_1 \mapsto m_1, \dots, \xi'_n \mapsto m_n
 \end{array}$$

Sur la base β , la notion de base interne β_c d'une pré-chronique c est définie inductivement par

$$\begin{array}{l}
 - \beta_\epsilon = \begin{cases} \vdash \xi_1, \dots, \xi_n; & \text{si } \beta = \vdash \xi_1, \dots, \xi_n \\ \xi \vdash \xi_1, \dots, \xi_n; & \text{si } \beta = \xi \vdash \xi_1, \dots, \xi_n \end{cases} \\
 - \beta_{c.(+,\xi,I)} = \xi.j \vdash \Gamma; \Delta_m \text{ pour un certain } j \in I \\
 \text{où } \beta_c = \vdash \Gamma; \Delta_m.
 \end{array}$$

Cette règle est non-déterministe car on ne sait pas à priori sur quel lieu va focaliser l'action suivante.

$$\begin{array}{l}
 - \beta_{c.(-,\xi,I)} = \vdash \xi_1, \dots, \xi_k; \left(\xi'_i \mapsto \begin{cases} m_i + 1 & \text{si } \exists i \in I, \xi'_i = \xi.i \\ m_i & \text{sinon} \end{cases} \right) \\
 \text{où } \beta_c = \xi \vdash \xi_1, \dots, \xi_k; (\xi'_i \mapsto m_i) \text{ avec la notation } \xi'_i \mapsto 0 \text{ si } \xi'_i \text{ n'appartient pas} \\
 \text{à la base interne.}
 \end{array}$$

La **base interne canoniquement associée** à une base β est β_ϵ que l'on identifie à β .

La multiplicité pour les lieux positifs indiquent le pointeur maximal utilisable.

Par exemple, sur la base $\vdash \xi$, la base interne de la pré-chronique

$$\begin{array}{c}
 (-, \xi.0, \{1; 2; 3\}) \\
 | \\
 (+, \xi, \{0\}) \\
 | \\
 (-, \xi.0, \{1; 2\}) \\
 | \\
 (+, \xi, \{0\})
 \end{array}$$

est $\vdash \xi$; $\xi.0.1 \mapsto 2$, $\xi.0.2 \mapsto 2$, $\xi.0.3 \mapsto 1$ et les pointeurs possibles sont

$$\xi; 0 : \xi.0.1; 1 : \xi.0.1; 0 : \xi.0.2; 1 : \xi.0.2; 0 : \xi.0.3$$

Les desseins universels sont définis inductivement sur les bases internes.

Comme en ludique probabiliste, pour la définition des desseins universels, un scalaire est noté $?$ et I_n décrit une énumération injective de $\mathcal{P}_{fn}(\mathbb{N})$.

Définition 8.27 – Pointeurs associés à une base interne positive

L'ensemble des pointeurs associés à une base interne positive

est

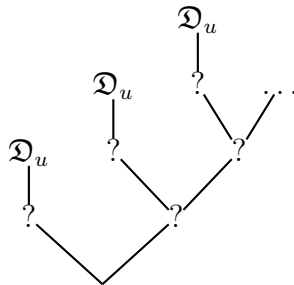
$$\vdash \xi_1, \dots, \xi_k; \xi'_1 \mapsto m_1, \dots, \xi'_n \mapsto m_n$$

$$\{\xi_1; \dots; \xi_k\} \cup \bigcup_{j=1}^n \{p : \xi'_j \mid p < m_j\}$$

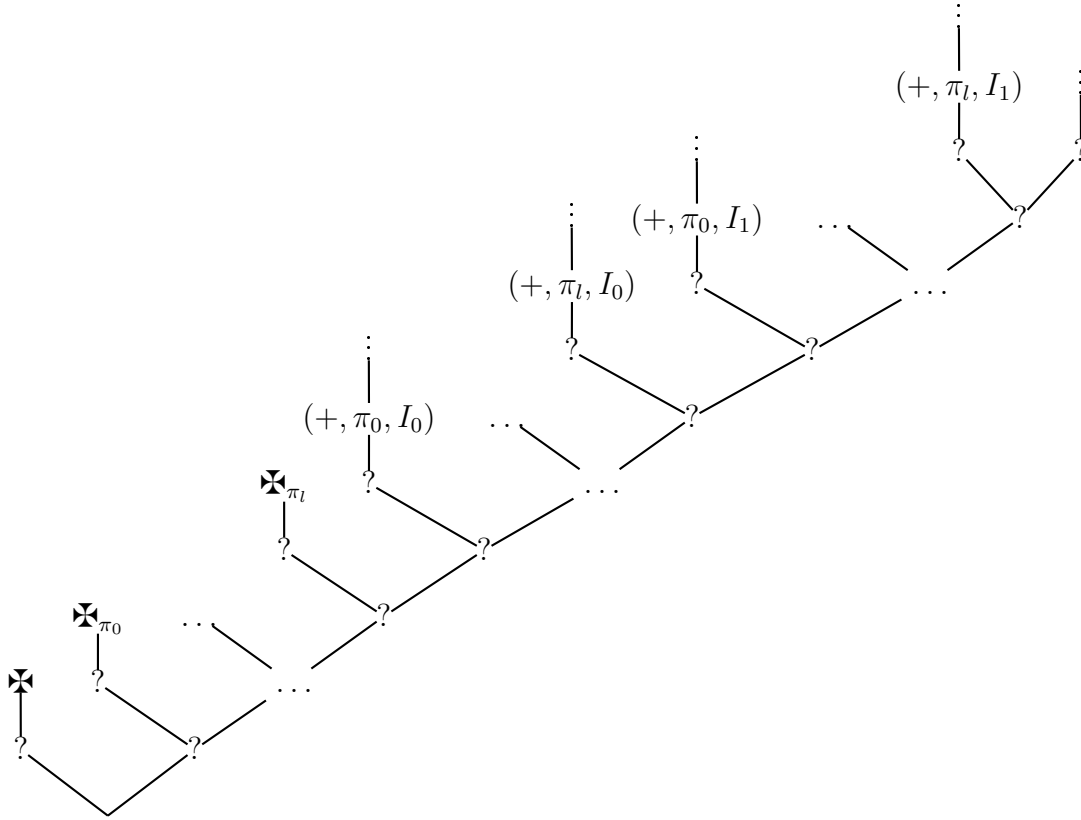
En notant $?$ un scalaire pour le moment non défini et par (I_n) une énumération injective de $\mathcal{P}_{fn}(\mathbb{N})$, le dessein universel de base interne

$$\vdash \xi_1, \dots, \xi_k; \xi'_1 \mapsto m_1, \dots, \xi'_n \mapsto m_n$$

est



où \mathcal{D}_u est le dessein suivant



où $\{\pi_1, \dots, \pi_l\}$ est l'ensemble des pointeurs associés à la base interne

$$\vdash \xi_1, \dots, \xi_k; \xi'_1 \mapsto m_1, \dots, \xi'_n \mapsto m_n$$

avec

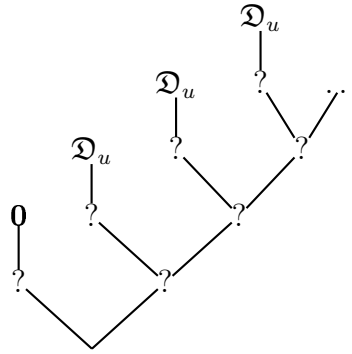
$$(+, \xi, \{i_1, \dots, i_n\}) = \mathcal{UD}_{\xi, i_1 \vdash \Gamma; \Delta}(\alpha) \cdot \dots \cdot \mathcal{UD}_{\xi, i_n \vdash \Gamma; \Delta}(\alpha)$$

et

$$(+, p : \xi, \{i_1, \dots, i_n\}) = \mathcal{UD}_{\xi, i_1 \vdash \Gamma; \Delta}(\alpha) \cdot \dots \cdot \mathcal{UD}_{\xi, i_n \vdash \Gamma; \Delta}(\alpha)$$

Contrairement au cas de la ludique probabiliste (page 106), on n'a pas à faire attention aux contextes car il n'y a plus de contraintes de linéarité.

Le dessin universel sur la base $\xi \vdash \Gamma; \Delta$ est le dessin



où \mathcal{D}_u est le dessein suivant :

$$\mathcal{U}\mathcal{D}_{\vdash\Gamma;\Delta(\xi,\{j_1,\dots,j_n\})}(\alpha) \quad \mathcal{U}\mathcal{D}_{\vdash\Gamma;\Delta(\xi,\{l_1,\dots,l_m\})}(\alpha)$$

$$(-, \xi, \{j_1, \dots, j_n\}) \quad \dots \quad (-, \xi, \{l_1, \dots, l_m\})$$

où $\Delta(\xi, I) = \left(\xi'_i \mapsto \begin{cases} m_i + 1 & \text{si } \exists i \in I, \xi'_i = \xi.i \\ m_i & \text{sinon} \end{cases} \right)$ et $\Delta = (\xi'_i \mapsto m_i)$ en utilisant la notation $\xi'_i \mapsto 0$ si ξ'_i n'appartient pas à la base interne.

Il reste à décrire les scalaires qui apparaissent dans le dessein universel. Une union dénombrable d'ensembles dénombrables étant dénombrable, les positions pour les scalaires ? sont dénombrables. On se donne une énumération bijective $p = (p_i)_{i \in \mathbb{N}}$ de ces positions. Le scalaire ? est $2^{-(i+1)}\alpha$ où i est l'indice dans p de la position du scalaire ?.

Définition 8.28 – Dessein universel

Soit β une base. Le **dessein universel** $\mathcal{U}\mathcal{D}_\beta(\alpha)$ est le dessein que l'on vient de décrire.

Remarque 8.3 Les desseins universels sont bien des desseins. De plus, les scalaires d'un dessein universel sont tous différents et leur somme totale est bornée par 1. Ainsi, toutes les sommes partielles de scalaires sont bornées par 1. L'adjectif universel provient de l'observation que tous les desseins avec pointeurs de base β sont inclus dans le dessein universel de base β au sens des projections sans coefficients (définition 8.4 page 209) : l'ensemble des sous-desseins des projections sans coefficients du dessein universel de base β est l'ensemble des desseins avec pointeurs de base β .

Proposition 8.5 – Scalaires dans $[\mathcal{D}, \mathcal{U}\mathcal{D}_{\beta^\perp}(\alpha)]$

Soit \mathcal{D} un dessein de base unaire β . Pour tout α , on a

$$\text{Scal}(\mathcal{D}) \subseteq \text{Scal}([\mathcal{D}, \mathcal{U}\mathcal{D}_{\beta^\perp}(\alpha)]) \subseteq \text{Scal}(\mathcal{D}) \cup \{2^{-n}\alpha \mid n \in \mathbb{N}\}$$

Démonstration : On procède comme en ludique probabiliste.

La preuve repose sur plusieurs remarques :

- Après une action négative, le dessein universel $\mathcal{U}\mathcal{D}_{\beta^\perp}(\alpha)$ utilise (entre autres actions) un démon et un démon focalisé sur tous les lieux (positifs) de la base interne (locale).
- Après une action négative, le dessein universel $\mathcal{U}\mathcal{D}_{\beta^\perp}(\alpha)$ essaie de focaliser sur toutes les actions possibles.
- Les actions positives sont suivies de toutes les actions négatives possibles.

Ce qui implique (par récurrence) que tout nœud de \mathcal{D} est visité lors de la normalisation du réseau $\mathcal{D}, \mathcal{U}\mathcal{D}_{\beta^\perp}(\alpha)$ et que le coefficient associé apparaît dans le résultat $\llbracket \mathcal{D}, \mathcal{U}\mathcal{D}_{\beta^\perp}(\alpha) \rrbracket$. ■

Remarque 8.4 *La proposition précédente implique que*

$$\text{Scal}^*(\mathcal{D}) = \text{Scal}^*(\llbracket \mathcal{D}, \mathcal{U}\mathcal{D}_{\beta^\perp}(1) \rrbracket)$$

Proposition 8.6 – Existence de $\alpha_{\text{Scal}^*(\mathcal{D})}$

Soit \mathcal{D} un dessein de base unaire β . Il existe un scalaire $\alpha_{\text{Scal}^*(\mathcal{D})} \in]0; 1[$ tel que

$$\text{Scal}^*(\mathcal{D}) \cap \alpha_{\text{Scal}^*(\mathcal{D})}\mathbb{Q} = \emptyset$$

Démonstration : On procède exactement comme en ludique probabiliste. ■

Définition 8.29 – Le scalaire $\alpha_{\mathcal{D}}$

Soit \mathcal{D} un dessein de base unaire. Soit $\alpha_{\mathcal{D}}$ un coefficient $\alpha_{\text{Scal}^*(\mathcal{D})}$ vérifiant la propriété de la proposition 8.6 ne dépendant que de $\text{Scal}^*(\mathcal{D})$.

Définition 8.30 – Opposant universel

Soit \mathcal{D} un dessein sur une base unaire β . L'**opposant universel** de \mathcal{D} est

$$\mathcal{U}\mathcal{D}_{\mathcal{D}} = \mathcal{U}\mathcal{D}_{\beta^\perp}(\alpha_{\llbracket \mathcal{D}, \mathcal{U}\mathcal{D}_{\beta^\perp}(1) \rrbracket})$$

Remarque 8.5 *Comme en ludique probabiliste, cette définition d'opposant universel est constructive dans le sens qu'elle dépend uniquement des résultats des normalisations de \mathcal{D} avec d'autres desseins et non de l'introspection de ses scalaires. Les résultats des normalisations de \mathcal{D} sont observables puisque ce sont fondamentalement des coefficients.*

Définition 8.31 – Scalaire maximal dans un coefficient

Un scalaire (identifié à son occurrence) est **maximal** dans un coefficient s'il est suivi par le coefficient 1.

La proposition clef pour la séparation est la suivante :

Proposition 8.7 – Injectivité de $\mathcal{D} \mapsto \llbracket \mathcal{D}, \mathcal{U}\mathcal{D}_{\mathcal{D}} \rrbracket$

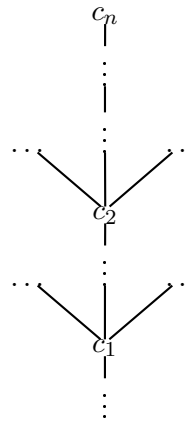
Sur une base unaire donnée, l'application $\mathcal{D} \mapsto \llbracket \mathcal{D}, \mathcal{U}\mathcal{D}_{\mathcal{D}} \rrbracket$ est injective.

Démonstration: On prouve par récurrence que les n premiers niveaux du dessein sont calculables à partir de $[\mathfrak{D}, \mathfrak{U}\mathfrak{D}]$.

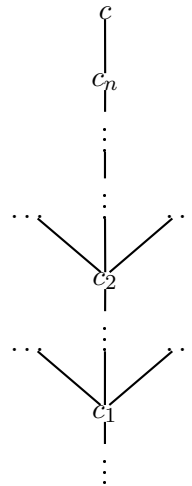
Pour passer du niveau n au niveau $n + 1$, la difficulté est de déterminer la valeur des actions aux feuilles des coefficients (qui se calculent aisément, comme en ludique probabiliste). Le problème est celui de la détermination des pointeurs.

L'idée fondamentale est de considérer uniquement les branches dans lesquelles chaque scalaire maximal du dessein universel intervient au plus une fois.

- ces branches suffisent pour décrire toutes les c -chroniques de \mathfrak{D} (c'est ici que servent les répétitions infinies des desseins \mathfrak{D}_u dans la définition des desseins universels négatifs);
- dans une telle branche, le calcul d'un pointeur est aisé. Le pointeur de l'action K_n de la c -chronique $c_1.K_1 \dots c_n.K_n$ est donné ainsi :
 - la pré-chronique $c_1.K_1 \dots c_n$ donne le coefficient



- la c -chronique $c_1.K_1 \dots c_n.K_n$ donne



- Si $K_n = (+, k : \xi.i, I)$ alors les scalaires de c correspondent à une certaine copie dans $\mathfrak{U}\mathfrak{D}$ de l'action duale $(-, \xi.i, I)$. Cette action $(-, \xi.i, I)$ suit une action positive $(+, \pi_\xi, J)$. Par induction sur la hauteur de la c -chronique considérée, l'action duale de $(+, \pi_\xi, J)$ dans \mathfrak{D} est bien déterminée ce qui permet de calculer le k de $K_n = (+, k : \xi.i, I)$

- Si $K_n = (+, \xi, I)$ ou $K_n = \mathfrak{X}_\xi$ alors il n'y a aucun pointeur à calculer.
- Si $K_n = \mathfrak{X}_{k;\xi,i}$ alors les scalaires de c correspondent à une certaine copie dans $\mathcal{U}\mathcal{D}$ d'un dessein négatif de base $\xi.i \vdash \Lambda$. Ce dessein négatif suit une action positive $(+, \pi_\xi, J)$. Par induction sur la hauteur de la c -chronique considérée, l'action duale de $(+, \pi_\xi, J)$ dans \mathcal{D} est bien déterminée ce qui permet de calculer le k de $K_n = \mathfrak{X}_{k;\xi,i}$. ■

Proposition 8.8 – Séparation

La ludique probabiliste est séparée : pour tous desseins \mathcal{D}_1 et \mathcal{D}_2 sur une même base unaire, on a :

$$\mathcal{D}_1 = \mathcal{D}_2 \Leftrightarrow \forall \mathfrak{E}, \llbracket \mathcal{D}_1, \mathfrak{E} \rrbracket = \llbracket \mathcal{D}_2, \mathfrak{E} \rrbracket$$

Démonstration : Le point important, c'est que pour tout dessein \mathcal{D} , l'opposant universel $\mathcal{U}\mathcal{D}$ ne dépend que des interactions de \mathcal{D} . Soit $\mathcal{D}_1 \neq \mathcal{D}_2$ deux desseins sur une même base unaire.

- Si $\text{Scal}^*(\mathcal{D}_1) \neq \text{Scal}^*(\mathcal{D}_2)$ alors $\llbracket \mathcal{D}_1, \mathcal{U}\mathcal{D}_{\beta^\perp}(1) \rrbracket \neq \llbracket \mathcal{D}_2, \mathcal{U}\mathcal{D}_{\beta^\perp}(1) \rrbracket$ d'après la proposition 8.5 page 222.
- Si $\text{Scal}^*(\mathcal{D}_1) = \text{Scal}^*(\mathcal{D}_2)$ alors $\mathcal{U}\mathcal{D}_{\mathcal{D}_1} = \mathcal{U}\mathcal{D}_{\mathcal{D}_2}$. La proposition 8.7 page 223 permet de conclure. ■

Remarque 8.6 Comme en ludique probabiliste, pour séparer deux desseins, il suffit donc de deux opposants : un premier pour calculer les scalaires et éventuellement un deuxième pour séparer les desseins si cela ne suffit pas.

Comme en ludique probabiliste, cette proposition peut être renforcée.

Proposition 8.9 – Séparation par des desseins universels fixés

Soit α et α' deux scalaires distincts dans $]0; 1]$. Soit \mathcal{D}_1 et \mathcal{D}_2 deux desseins de base unaire β .

Les desseins \mathcal{D}_1 et \mathcal{D}_2 sont égaux si et seulement si :

$$\llbracket \mathcal{D}_1, \mathcal{U}\mathcal{D}_{\beta^\perp}(\alpha) \rrbracket = \llbracket \mathcal{D}_2, \mathcal{U}\mathcal{D}_{\beta^\perp}(\alpha) \rrbracket \wedge \llbracket \mathcal{D}_1, \mathcal{U}\mathcal{D}_{\beta^\perp}(\alpha') \rrbracket = \llbracket \mathcal{D}_2, \mathcal{U}\mathcal{D}_{\beta^\perp}(\alpha') \rrbracket$$

Démonstration : L'argument est identique à celui de la proposition 3.22 page 108 de la ludique probabiliste. ■

8.5.3 Stabilité

Définition 8.32 – Compatibilité pleine

Deux desseins sont **pleinement compatibles** lorsqu'ils sont pleinement inclus dans un même dessein.

Proposition 8.10 – Inclusion et normalisation

Soit $\mathfrak{R} \subseteq \mathfrak{R}'$. On a

$$\llbracket \mathfrak{R} \rrbracket \subseteq \llbracket \mathfrak{R}' \rrbracket$$

Démonstration : On utilise le codage. Soient $\mathfrak{R}_1 \subseteq \mathfrak{R}'_1$ deux réseaux. Soient \mathfrak{R}_2 et \mathfrak{R}'_2 leurs traduits. Par monotonie de la traduction, on a $\mathfrak{R}_2 \subseteq \mathfrak{R}'_2$. Par le théorème de monotonie de la ludique avec pointeurs (proposition 7.8 page 194) $\llbracket \mathfrak{R}_2 \rrbracket \subseteq \llbracket \mathfrak{R}'_2 \rrbracket$ et donc $\llbracket \mathfrak{R}_1 \rrbracket \subseteq \llbracket \mathfrak{R}'_1 \rrbracket$ par monotonie de la traduction inverse. ■

L'inclusion pleine permet de définir une notion d'intersection : l'intersection pleine.

Proposition 8.11 – Intersection pleine

Pour toute famille (\mathfrak{D}_i) de desseins compatibles, l'ensemble formé des c -chroniques pleinement incluses dans chaque \mathfrak{D}_i décrit un dessein.

Démonstration : Soit X l'ensemble formé des c -chroniques pleinement incluses dans chaque \mathfrak{D}_i . Il suffit de remarquer que deux c -chroniques dans X sont cohérentes. On conclut par la proposition 3.12 page 98. ■

Définition 8.33 – Intersection pleine

L'**intersection pleine** d'une famille (\mathfrak{D}_i) de desseins compatibles $\bigcap_i \mathfrak{D}_i$ est le dessein formé des c -chroniques pleinement incluses dans chaque \mathfrak{D}_i .

Proposition 8.12 – Stabilité

Soit (\mathfrak{R}_i) une famille de réseaux inclus dans un réseau \mathfrak{R} . On a

$$\llbracket \bigcap \mathfrak{R}_i \rrbracket = \bigcap \llbracket \mathfrak{R}_i \rrbracket$$

Démonstration : On procède comme en ludique simple.

Il faut d'abord vérifier que cette définition a bien un sens c'est-à-dire que les desseins $\llbracket \mathfrak{R}_i \rrbracket$ sont bien compatibles. C'est vrai par la Proposition 8.10.

On utilise la normalisation par codage (définie section 8.3 page 210) ce qui permet de relever le théorème de stabilité de la ludique avec pointeurs au cas exponentiel. On ne peut directement utiliser la proposition 8.3 page 215 car cette proposition ne prend pas en compte l'intersection. ■

8.5.4 Monotonie**Proposition 8.13 – Monotonie**

La normalisation est croissante vis-à-vis de l'ordre extensionnel \preceq .

$$\forall i, \mathfrak{D}_i \preceq \mathfrak{D}'_i \implies \llbracket \mathfrak{D}_1, \dots, \mathfrak{D}_n \rrbracket \preceq \llbracket \mathfrak{D}'_1, \dots, \mathfrak{D}'_n \rrbracket$$

Démonstration : On le vérifie par traduction en ludique avec pointeurs : l'ordre extensionnel commute avec la traduction. ■

8.6 Comportements

Les comportements et connecteurs de la ludique exponentielle généralisent à la fois les comportements et connecteurs de la ludique probabiliste et de la ludique avec pointeurs. Les notions d'incarnation, d'éthique complète ou de combinaison linéaire sont adaptées.

8.6.1 Définitions

Comme en ludique probabiliste, un dessein a un orthogonal non vide si et seulement si il est immédiatement total (définition 8.2 page 208). La notion d'éthique exponentielle est donc restreinte aux desseins immédiatement totaux.

Définition 8.34 – Éthique

Une **éthique** est un ensemble de desseins immédiatement totaux de même base.

Définition 8.35 – Orthogonal d'une éthique

L'**orthogonal** d'une éthique est défini par

$$E^\perp = \{\mathfrak{F} \mid \forall \mathcal{D} \in E, \mathcal{W}(\llbracket \mathfrak{F}, \mathcal{D} \rrbracket) = 1\}$$

Définition 8.36 – Comportement multiple

Un **comportement multiple** est une éthique égale à son bi-orthogonal.

Définition 8.37 – Ensemble des desseins linéaires sur la base

Un dessein \mathcal{D} sur une base β est **linéaire** si ses projections sans coefficients sont linéaires.

Soit E une éthique sur une base β . On note $\text{Lin}(E)$ l'**ensemble des desseins de E linéaires sur β** .

Définition 8.38 – Comportement linéaire

Un **comportement linéaire** est une éthique G telle que

$$G = \text{Lin}(G^{\perp\perp})$$

L'incarnation est définie de manière similaire.

Proposition 8.14 – Incarnation

Pour tout comportement multiple ou linéaire G , et tout dessein \mathcal{D} de G , il existe un unique dessein $\mathcal{D}_0 \subseteq \mathcal{D}$ de G tel que

$$\forall \mathfrak{E} \in G, \mathfrak{E} \subseteq \mathcal{D} \Rightarrow \mathcal{D}_0 \subseteq \mathfrak{E}$$

De plus, ce dessein \mathcal{D}_0 se calcule en annulant les coefficients positifs de \mathcal{D} non visités par un dessein de G^\perp . Le dessein \mathcal{D}_0 est l'intersection des desseins de G inclus dans \mathcal{D} .

Démonstration : On procède comme en ludique simple en utilisant la stabilité (proposition 8.12 page 226). ■

Définition 8.39 – Incarnation

L'**incarnation** d'un dessein \mathcal{D} dans un comportement G est le dessein de la proposition 8.14.

$$|\mathcal{D}|_G = \bigcap \{ \mathfrak{E} \in G \mid \mathfrak{E} \subseteq \mathcal{D} \}$$

Comme en ludique simple, on utilise deux notions d'éthique complète.

Définition 8.40 – Éthique multiple complète

Une éthique est **une éthique multiple complète** si $\text{Pss}(|E^{\perp\perp}|) \subseteq E$.

Définition 8.41 – Éthique linéaire complète

Une éthique est **une éthique linéaire complète** si :

$$\text{Pss}(\text{Lin}(|E^{\perp\perp}|)) \subseteq \text{Lin}(E) = E$$

Remarque 8.7 La notion d'éthique linéaire complète est une sorte de fusion des notions correspondantes en ludique probabiliste et en ludique avec pointeurs.

En effet, en ludique probabiliste, une éthique complète vérifie :

$$\text{Pss}(|E^{\perp\perp}|) \subseteq E$$

et en ludique avec pointeurs, une éthique linéaire complète vérifie :

$$\text{Lin}(|E^{\perp\perp}|) \subseteq \text{Lin}(E) = E$$

Définition 8.42 – Bi-orthogonal linéaire

Le **bi-orthogonal linéaire** d'une éthique E est

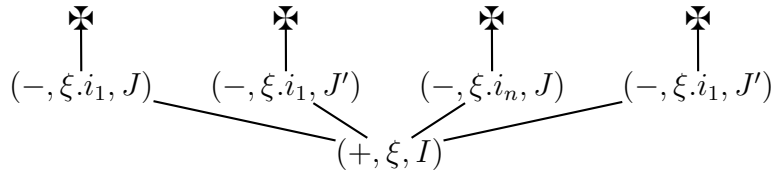
$$\text{Lin}(E^{\perp\perp})$$

8.6.2 Notions pour la complétude interne

On se contente de passer en revue les notions attachées au concept de répertoire.

Définition 8.43 – Répertoire d'un comportement

- Le **répertoire** d'un comportement positif G est l'ensemble des ramifications I telles que le dessein suivant appartienne à G



où $\begin{cases} I = \{i_1, \dots, i_n\} \\ J, J' \text{ parcourent } \mathcal{P}_{fin}(\mathbb{N}) \end{cases}$

- Le **répertoire** d'un comportement négatif est l'ensemble des ramifications I

telles que le dessein $\begin{matrix} \otimes \\ \uparrow \\ (-, \xi, I) \end{matrix}$ est inclus dans l'incarnation dans G de

$\mathcal{D}ai^- = \begin{matrix} \otimes \\ \uparrow \\ (-, \xi, J) \end{matrix} \dots \begin{matrix} \otimes \\ \uparrow \\ (-, \xi, J') \end{matrix}$ où J, J' parcourent $\mathcal{P}_{fin}(\mathbb{N})$.

- le **répertoire** d'une éthique est le répertoire de son bi-orthogonal.

Proposition 8.15 – Caractérisation du répertoire d'un comportement positif

Le répertoire d'un comportement multiple ou linéaire positif G sur la base $\vdash \xi$ est :

1. l'ensemble des ramifications I tel qu'un dessein de G commence par $(+, \xi, I)$;
2. l'ensemble des ramifications I tel qu'un dessein de G a une action $(+, \xi, I)$ visitée par G^\perp .

Démonstration : C'est un mélange des preuves pour la ludique probabiliste et la ludique avec pointeurs. ■

Naturellement, la conservation par orthogonalité du répertoire est encore vérifiée :

Proposition 8.16 – Répertoire

Tout comportement sur une base unaire a le même répertoire que son orthogonal.

Définition 8.44 – Comportements disjoints et connexes

- Deux comportements sont **disjoints** si leurs répertoires sont disjoints.
- Une éthique E est **connexe** si son répertoire est un singleton $\{I\}$. L'ensemble I est appelé la **ramification** de E .

Proposition 8.17 – Caractérisation des comportements disjoints

- Deux comportements positifs sont disjoints si et seulement si leur intersection est réduite aux combinaisons linéaires de poids 1 de démons focalisés.
- Deux comportements négatifs sont disjoints si et seulement si

$$\forall \mathfrak{D} \in \text{Pss}(G), \forall \mathfrak{E} \in \text{Pss}(H), |\mathfrak{D}|_G \cap |\mathfrak{E}|_H = \emptyset.$$

Démonstration : On réutilise la preuve en ludique probabiliste (proposition 5.19 page 144). ■

Comme pour toutes les variantes de la ludique vues jusqu'ici, les propositions suivantes permettent de prouver la complétude interne pour le *tenseur*.

Proposition 8.18 – Existence d'une projection

Soit $\mathbb{M} \subseteq \mathbb{N}$ un réservoir. Tout dessein positif semi-simple $\mathfrak{D} = (+, \xi, I). \mathfrak{R}$ est égal à $\mathfrak{D}_{\mathbb{M}} \otimes \mathfrak{D}_{\mathbb{N} \setminus \mathbb{M}}$ où $\mathfrak{D}_{\mathbb{M}}$ commence par $(+, \xi, I \cap \mathbb{M})$ et $\mathfrak{D}_{\mathbb{N} \setminus \mathbb{M}}$ commence par $(+, \xi, I \setminus \mathbb{M})$. Le dessein $\mathfrak{D}_{\mathbb{M}}$ est le **projeté** de \mathfrak{D} sur \mathbb{M} .
On étend cette définition en posant $\mathfrak{X}_{\xi_{\mathbb{M}}} = \mathfrak{X}_{\xi}$.

Proposition 8.19 – Projection

Soit E une éthique connexe et \mathbb{M} un réservoir. On a $(E^{\perp\perp})_{\mathbb{M}} = (E_{\mathbb{M}})^{\perp\perp}$.

Définition 8.45 – Indépendance

Deux répertoires R_1 et R_2 sont **indépendants** si

$$\forall I_1, J_1 \in R_1, I_2, J_2 \in R_2, I_1 \cup I_2 = J_1 \cup J_2 \Rightarrow I_1 = J_1, I_2 = J_2$$

Deux comportements sont **indépendants** si leurs répertoires sont indépendants.

Définition 8.46 – Comportements étrangers

Deux comportements sont **étrangers** si leurs réservoirs sont disjoints.

8.7 Connecteurs

Les connecteurs sont définis comme pour la ludique probabiliste et la ludique avec pointeurs. En particulier, les connecteurs exponentiels sont définis de manière similaire.

Les connecteurs habituels portent sur les comportements linéaires. Le \flat et le \sharp font passer des comportements linéaires aux comportements multiples.

8.7.1 Décalages

Définition 8.47 – Décalage

– Si \mathcal{D} est un dessein positif de base $\vdash \xi.i$ alors

$$\uparrow\mathcal{D} = \begin{cases} (-, \xi, \{i\}).\mathcal{D} & \text{si } \mathcal{D} \neq \text{fid} \\ \text{Gfunt} & \text{sinon} \end{cases}$$

– Si \mathcal{D} est un dessein négatif de base $\xi.i \vdash$ alors

$$\downarrow\mathcal{D} = (+, \xi, \{i\}).\mathcal{D}$$

– Sur une base unaire, le **décalage** d'un comportement linéaire G est le bi-orthogonal linéaire des décalages de ses desseins. Lorsque G est positif de base $\vdash \xi.i$ (resp. négatif de base $\xi.i \vdash$), on le note $\uparrow G$ (resp. $\downarrow G$).

Comme pour la ludique avec pointeurs, l'éthique complète pour le décalage négatif (d'un comportement positif) fait intervenir un bi-orthogonal.

Proposition 8.20 – Complétude linéaire interne pour les décalages

Soit G un comportement linéaire.

- Si G est positif, alors $\{\uparrow\mathcal{D} \mid \mathcal{D} \in G^{\perp\perp}\}$ est une éthique complète pour $\uparrow G$.
- Si G est négatif, alors $\downarrow G = \{\downarrow\mathcal{D} \mid \mathcal{D} \in G\} \cup \{\mathbf{X}_\xi\}$.

Démonstration : La preuve est une fusion des preuves en ludique probabiliste et en ludique avec pointeurs. ■

8.7.2 Exponentielles

On reprend les constructions de la ludique avec pointeurs.

Définition 8.48 – Exponentielles

- Si G est un comportement linéaire positif, alors $?G$ est le bi-orthogonal linéaire des décalages négatifs des desseins du bi-orthogonal de G .
- Si G est un comportement linéaire négatif, alors $!G$ est le bi-orthogonal linéaire des décalages positifs des desseins de G .

Définition 8.49 – Les connecteurs bémol \flat et dièse \sharp

- Si G est un comportement linéaire positif, on note $\flat G$ son bi-orthogonal.
- Si G est un comportement linéaire négatif, on note $\sharp G$ son bi-orthogonal (égal à G).

Proposition 8.21 – Complétude linéaire interne pour les exponentielles

Soit G un comportement linéaire.

- Si G est positif, alors $\{\uparrow\mathcal{D} \mid \mathcal{D} \in \flat G\}$ est une éthique complète pour $?G$.
- Si G est négatif, alors $!G = \{\downarrow\mathcal{D} \mid \mathcal{D} \in \sharp G\} \cup \{\mathbf{X}_\xi\} = \{\downarrow\mathcal{D} \mid \mathcal{D} \in G\} \cup \{\mathbf{X}_\xi\}$.

La définition des exponentielles est à nouveau identique à celle des décalages. Cela est corrigé avec l'uniformité à la section 8.8.4 page 236.

8.7.3 Additifs

Définition 8.50 – Le connecteur \bigcap

Soit (G_k) une famille de comportements multiples (resp. linéaires) sur une même base. On définit le comportement multiple (resp. linéaire) $\bigcap_k G_k$ par :

$$\bigcap_k G_k = \{\mathfrak{D} \mid \forall k, \mathfrak{D} \in G_k\}$$

Définition 8.51 – Le connecteur $\bigcup^{\perp\perp}$

Soit (G_k) une famille de comportements multiples (resp. linéaires) sur une même base. On définit le comportement multiple (resp. linéaire) $\bigcup_k^{\perp\perp} G_k$ par :

$$\bigcup_k^{\perp\perp} G_k = \{\mathfrak{D} \mid \exists k, \mathfrak{D} \in G_k\}^{\perp\perp} \text{ (resp. } \bigcup_k^{\perp\perp} G_k = \text{Lin}(\{\mathfrak{D} \mid \exists k, \mathfrak{D} \in G_k\}^{\perp\perp}))$$

Définition 8.52 – Le connecteur avec $\&$

Le connecteur $\&$ est le connecteur \bigcap appliqué à des comportements négatifs disjoints.

Le connecteur $\&$ est encore un produit cartésien sur les incarnations :

Proposition 8.22 – Mystère de l'incarnation

Pour tous comportements disjoints G et H sur une même base négative,

$$\text{Pss}(|G \& H|) = \text{Pss}(|G|) \times \text{Pss}(|H|)$$

Définition 8.53 – Le connecteur *plus* \oplus

Le connecteur \oplus est le connecteur $\bigcup^{\perp\perp}$ appliqué à des comportements linéaires positifs disjoints.

Proposition 8.23 – Propriété de la disjonction

Pour tous comportements linéaires positifs disjoints,

$$\text{Pss}(G \oplus H) = \text{Pss}(G) \cup \text{Pss}(H)$$

8.7.4 Multiplicatifs

Comme en ludique probabiliste, on définit le tenseur par bilinéarité.

Définition 8.54 – Le tenseur

Les différentes versions du tenseur sont définies par bilinéarité :

- Le produit tensoriel de deux comportements linéaires positifs est l'ensemble des combinaisons linéaires des produits tensoriels des parties semi-simples

$$\text{Pss}(G \otimes H) = \text{Pss}(G) \otimes \text{Pss}(H)$$

et donc

$$G \otimes H = \text{Cl}(\text{Pss}(G) \otimes \text{Pss}(H))$$

- Pour des desseins semi-simples, la construction est similaire à la construction habituelle (ludique simple, probabiliste ou avec pointeurs).

Définition 8.55 – Le connecteur \bowtie

Le nœud papillon \bowtie est le dual du tenseur commutatif maximal.

Proposition 8.24 – Complétude interne pour le tenseur

Si G_1 et G_2 sont positifs et indépendants alors $\{\mathcal{D}_1 \circledast \mathcal{D}_2 \mid \mathcal{D}_1 \in G_1, \mathcal{D}_2 \in G_2\}$ et $\{\mathcal{D}_1 \circledast \mathcal{D}_2 \mid \mathcal{D}_1 \in \text{Pss}(G_1), \mathcal{D}_2 \in \text{Pss}(G_2)\}$ sont des éthiques complètes pour $G_1 \circledast G_2$.

Démonstration : Comme pour les autres versions de la ludique, on utilise la proposition d'existence de projection (proposition 8.18 page 230). ■

8.8 Uniformité

On utilise une technique similaire à celle employée pour la ludique probabiliste (chapitre 6 page 149).

8.8.1 Bi-comportements

On reprend les notions de ludique probabiliste.

Définition 8.56 – Dessein partiel

Un *dessein partiel d'une éthique* est un dessein pleinement inclus dans un dessein de cette éthique.

La définition suivante permet de parler de la partie d'un coefficient qui précède un ensemble d'actions.

Définition 8.57 – Coefficient sous-jacent à un ensemble d'actions

Le **coefficient sous-jacent** à un ensemble d'actions X dans un coefficient c est le coefficient c' obtenu en annulant les feuilles vides de c dont l'action n'est pas dans X .

Si \mathcal{D} est un dessein sur la base $\vdash \xi$, on note $\text{Coef}_{\mathfrak{X}, \mathfrak{X}_\xi}(\mathcal{D})$ le coefficient sous-jacent au premier coefficient (le coefficient à la racine) de \mathcal{D} de la paire $\{\mathfrak{X}; \mathfrak{X}_\xi\}$. De même, on note $\text{Coef}_{pr}(\mathcal{D})$ le coefficient sous-jacent au premier coefficient de \mathcal{D} de l'ensemble des actions propres. Le premier coefficient, correspondant à l'ensemble des actions est noté $\text{Coef}_{fst}(\mathcal{D})$.

Définition 8.58 – Bi-éthique

Une **bi-éthique** est une paire (E, d) telle que :

- E est une éthique.
- d est une pré-distance sur les desseins partiels de E telle que :
 - Cas positif.
 - $d(\mathfrak{F}id, \mathfrak{F}id) = 0$
 - $d(\mathfrak{X}_\xi, \mathfrak{X}_\xi) = 0$
 - $\forall \mathcal{D}_1, \mathcal{D}_2 \in E^p, d(\mathcal{D}_1, \mathcal{D}_2) \geq \delta_c(\text{Coef}_{fst}(\mathcal{D}_1), \text{Coef}_{fst}(\mathcal{D}_2))$
 - $\forall \mathcal{D}_1, \mathcal{D}_2 \in E^p, d(\mathcal{D}_1, \mathcal{D}_2) \geq \delta_c(\text{Coef}_{\mathfrak{X}, \mathfrak{X}_\xi}(\mathcal{D}_1), \text{Coef}_{\mathfrak{X}, \mathfrak{X}_\xi}(\mathcal{D}_2))$
 - Cas négatif.
 - $\forall \mathcal{D}_1, \mathcal{D}_2 \in E^p, d(\mathcal{D}_1, \mathcal{D}_2) \geq \delta_c(\text{Coef}_{fst}(\mathcal{D}_1), \text{Coef}_{fst}(\mathcal{D}_2))$
 - $d(\mathfrak{D}ai^-, \mathfrak{D}ai^-) = 0$
 - $d(\mathfrak{G}funk, \mathfrak{G}funk) = 0$

Dans la définition suivante, on note $\text{Semi-Simp}(E)$ les desseins semi-simples d'un ensemble de desseins E .

Définition 8.59 – Orthogonal d'une bi-éthique

L'**orthogonal** d'une bi-éthique (E, d) sur une base unaire est la paire (E^\perp, d^\perp) où la pré-distance d^\perp est définie par

$$\forall \mathfrak{E}_1, \mathfrak{E}_2 \in E^{\perp p}, d^\perp(\mathfrak{E}_1, \mathfrak{E}_2) = \sup_{\mathcal{D}_1, \mathcal{D}_2 \in \text{Semi-Simp}(E^p)} (\delta_c(\llbracket \mathcal{D}_1, \mathfrak{E}_1 \rrbracket, \llbracket \mathcal{D}_2, \mathfrak{E}_2 \rrbracket) - d(\mathcal{D}_1, \mathcal{D}_2))$$

Remarque 8.8 La définition de l'orthogonal d'une bi-éthique en ludique probabiliste (définition 6.7 page 154) utilise simplement les desseins simples car le but était de prouver rapidement un résultat de complétude forte.

La définition utilisée ici considère les desseins semi-simples car cela paraît plus naturel et la recherche éventuelle d'un théorème de complétude ne saurait utiliser les techniques de la deuxième partie à cause des exponentielles : l'utilisation de nouveaux connecteurs implique, au minimum, de refaire les preuves de la ludique simple et non de les utiliser en factorisant les résultats pour la complétude forte comme pour la ludique probabiliste (section 6.5 page 165).

Proposition 8.25 – Orthogonal d’une bi-éthique

| *L’orthogonal d’une bi-éthique est une bi-éthique.*

Démonstration : La preuve est similaire à celle de la proposition correspondante en ludique probabiliste (proposition 6.3 page 154). ■

Définition 8.60 – Bi-comportement

| *Un **bi-comportement** est une bi-éthique égale à son bi-orthogonal.*

Proposition 8.26 – Orthogonal d’une bi-éthique et bi-comportement

| *L’orthogonal d’une bi-éthique est un bi-comportement.*

| *De plus, pour toute bi-éthique (E, d) et tous desseins semi-simples \mathfrak{D}_1 et \mathfrak{D}_2 de E ,*

$$d^{\perp\perp}(\mathfrak{D}_1, \mathfrak{D}_2) \leq d(\mathfrak{D}_1, \mathfrak{D}_2)$$

Démonstration : On procède comme pour la proposition 6.4 page 156. ■

Définition 8.61 – Dessein uniforme

| *Un dessein \mathfrak{D} est **uniforme** dans un bi-comportement (G, d) si*

$$d(\mathfrak{D}, \mathfrak{D}) = 0$$

8.8.2 Connecteurs

L’uniformité sur les connecteurs de la logique linéaire est définie comme en ludique probabiliste. La nouveauté réside dans la gestion des décalages qui permet de les différencier des exponentielles.

8.8.3 Décalages

Le décalage négatif d’un comportement linéaire fait apparaître des desseins qui sont des décalages de desseins non linéaires. Pour “enlever” par uniformité ces desseins des bi-comportements, on utilise la technique décrite dans la définition suivante.

Définition 8.62 – Décalages

- Le **décalage négatif** d'un bi-comportement linéaire positif de base $\vdash \xi.i$ est

$$\uparrow(G, d) = (\uparrow G, \uparrow d) = (\uparrow G, \uparrow d)^{\perp\perp}$$

où

$$\uparrow d(\dots \uparrow \mathcal{D}_1 \dots, \dots \uparrow \mathcal{D}_2 \dots) = \begin{cases} d(\mathcal{D}_1, \mathcal{D}_2) & \text{si } \mathcal{D}_1 \text{ et } \mathcal{D}_2 \text{ sont linéaires} \\ 1 & \text{sinon} \end{cases}$$

et $\dots \uparrow \mathcal{D}_k \dots$ est une notation pour un dessein négatif dont la continuation de $(-, \xi, \{i\})$ est \mathcal{D}_k .

- Le **décalage positif** d'un bi-comportement négatif de base $\xi.i \vdash$ est

$$\downarrow(G, d) = (\downarrow G, \downarrow d) = (\uparrow(G^\perp, d^\perp))^\perp$$

Remarque 8.9 Si \mathcal{D}_1 ou \mathcal{D}_2 n'est pas linéaire alors $\uparrow d(\dots \uparrow \mathcal{D}_1 \dots, \dots \uparrow \mathcal{D}_2 \dots) = 1$.

Cela implique que les desseins non-linéaires n'interviennent pas dans le calcul de la distance pour $\downarrow G^\perp$ et donc pour $\uparrow G$.

Cependant, les conséquences logiques de cette définition ne sont pas claires (pour la recherche d'un éventuel théorème de complétude par exemple).

8.8.4 Exponentielles

Contrairement au décalage négatif, le *pourquoi pas* d'un comportement linéaire positif "autorise" les décalages de desseins non linéaires ce qui explique les différences entre la définition suivante et la définition 8.62.

Définition 8.63 – Exponentielles

- Le *pourquoi pas* d'un bi-comportement linéaire positif de base $\vdash \xi.i$ est

$$?(G, d) = (?G, ?d) = (?G, ?d)^{\perp\perp}$$

où

$$?d(\dots \uparrow \mathcal{D}_1 \dots, \dots \uparrow \mathcal{D}_2 \dots) = d(\mathcal{D}_1, \mathcal{D}_2)$$

et $\dots \uparrow \mathcal{D}_k \dots$ est une notation pour un dessein négatif dont la continuation de $(-, \xi, \{i\})$ est \mathcal{D}_k .

- Le *bien sûr* d'un bi-comportement négatif de base $\xi.i \vdash$ est

$$!(G, d) = (!G, !d) = (!G, !d)^{\perp\perp}$$

Comme pour les décalages en ludique probabiliste, les distances $!d$ et $?d$ sont complètes aux sens suivant de la proposition suivante.

Proposition 8.27 – Complétude interne pour les exponentielles

Pour tout bi-comportement négatif (G, d)

$$(!G, !d) = (!G, !'d)^{\perp\perp} = (!G, !'d)$$

Pour tout bi-comportement linéaire positif (G, d)

$$(?G, ?d) = (?G, ?'d)^{\perp\perp} = (?G, ?'d)$$

Démonstration : On réutilise la technique par orthogonalité ($'d$ et $!d$ sont orthogonales l'une de l'autre). ■

8.8.5 Additifs

Les additifs sont définis sur les comportements linéaires comme en ludique probabiliste.

Définition 8.64 – Intersection

L'intersection de bi-comportements négatifs de même base est

$$\cap_k (G_k, d_k) = (\cap_k G_k, \cap_k d_k)$$

avec

$$\cap_k d_k = \sup d_k$$

Définition 8.65 – Union

L'union de bi-comportements positifs linéaires de même base est

$$\cup_k^* (G_k, d_k) = (\cup_k^* G_k, \cup_k^* d_k) = (\cup_k^* G_k, \cup_k d_k)^{\perp\perp}$$

avec

$$\cup_k d_k = \inf d_k$$

Ces définitions (*intersection* et *union*) définissent les connecteurs *avec* et *plus* sur les comportements disjoints.

8.8.6 Multiplicatifs

Les multiplicatifs sont définis sur les comportements linéaires de manière similaire à la ludique probabiliste.

Définition 8.66 – Le nœud papillon

Le connecteur \bowtie de deux bi-comportements négatifs de même base (G_1, d_1) et (G_2, d_2) est :

$$(G_1, d_1) \bowtie (G_2, d_2) = (G_1 \bowtie G_2, d_1 \bowtie d_2)$$

avec

$$(d_1 \bowtie d_2)(\mathfrak{F}_1, \mathfrak{F}_2) = \sup d_2((\mathfrak{F}_1)\mathcal{D}_1, (\mathfrak{F}_2)\mathcal{D}_2) - d_1^\perp(\mathcal{D}_1, \mathcal{D}_2)$$

Proposition 8.28 – Le nœud papillon

Pour tous bi-comportements négatifs de même base (G_1, d_1) et (G_2, d_2) , la bi-éthique $(G_1 \bowtie G_2, d_1 \bowtie d_2)$ est un bi-comportement.

Démonstration : Comme en ludique probabiliste, il suffit (proposition 8.26 page 235) de montrer que c'est l'orthogonal d'une bi-éthique. C'est l'orthogonal des bi-éthiques $(G_1^\perp \otimes G_2^\perp, d')$ telles que sur les desseins semi-simples :

$$d'(\mathcal{D}_1^0, \mathcal{D}_2^0) = \min\{1; \inf_{\forall i \mathcal{D}_i^0 = \mathcal{D}_i \otimes \mathcal{D}'_i} d_1(\mathcal{D}_1, \mathcal{D}_2) + d_2(\mathcal{D}'_1, \mathcal{D}'_2)\}$$

En effet,

$$\begin{aligned} & (d_1 \bowtie d_2)(\mathfrak{F}_1, \mathfrak{F}_2) \\ &= \sup d_2((\mathfrak{F}_1)\mathcal{D}_1, (\mathfrak{F}_2)\mathcal{D}_2) - d_1^\perp(\mathcal{D}_1, \mathcal{D}_2) \\ &= \sup \delta_c(\llbracket (\mathfrak{F}_1)\mathcal{D}_1, \mathcal{D}'_1 \rrbracket, \llbracket (\mathfrak{F}_2)\mathcal{D}_2, \mathcal{D}'_2 \rrbracket) - d_2^\perp(\mathcal{D}'_1, \mathcal{D}'_2) - d_1^\perp(\mathcal{D}_1, \mathcal{D}_2) \\ &= \sup \delta_c(\llbracket \mathfrak{F}_1, \mathcal{D}_1 \otimes \mathcal{D}'_1 \rrbracket, \llbracket \mathfrak{F}_2, \mathcal{D}_2 \otimes \mathcal{D}'_2 \rrbracket) - (d_1^\perp(\mathcal{D}_1, \mathcal{D}_2) + d_2^\perp(\mathcal{D}'_1, \mathcal{D}'_2)) \\ &= \sup \delta_c(\llbracket \mathfrak{F}_1, \mathcal{D}_1 \otimes \mathcal{D}'_1 \rrbracket, \llbracket \mathfrak{F}_2, \mathcal{D}_2 \otimes \mathcal{D}'_2 \rrbracket) - d'(\mathcal{D}_1 \otimes \mathcal{D}'_1, \mathcal{D}_2 \otimes \mathcal{D}'_2) \end{aligned}$$

Il ne reste qu'à invoquer la surjectivité du tenseur sur les desseins semi-simples. ■

Remarque 8.10 On a utilisé le fait que la connaissance de la distance sur les desseins semi-simples suffit à définir l'orthogonal.

Comme pour la ludique probabiliste, on obtient la proposition suivante :

Proposition 8.29 – Distance et tenseur

Soient (G, d_G) et (H, d_H) deux bi-comportements positifs. On a pour tous desseins totaux, propres et simples :

$$d_{G \otimes H}(\mathcal{D}_1 \otimes \mathcal{D}_2, \mathcal{D}'_1 \otimes \mathcal{D}'_2) \leq \min\{1; d_G(\mathcal{D}_1, \mathcal{D}'_1) + d_H(\mathcal{D}_2, \mathcal{D}'_2)\}$$

8.9 Un mot sur l'interprétation

Les règles considérées pour les connecteurs exponentiels sont les suivantes :

$$\begin{array}{ll}
 \text{promotion} & \frac{\Gamma, P \vdash \Delta;}{\Gamma \vdash \Delta; !P^\perp} \qquad \text{déréliction} \quad \frac{\Gamma \vdash \Delta; P}{\Gamma, !P^\perp \vdash \Delta;} \\
 \text{affaiblissement} & \frac{\Gamma \vdash \Delta;}{\Gamma, !P^\perp \vdash \Delta;} \qquad \text{contraction} \quad \frac{\Gamma, !P^\perp, !P^\perp \vdash \Delta;}{\Gamma, !P^\perp \vdash \Delta;}
 \end{array}$$

La traduction des règles exponentielles de promotion et de dérélliction en ludique exponentielles se fait comme pour les traductions des règles de décalages [Gir01]. Les règles d'affaiblissement et de contraction ne sont pas observables en ludique et sont donc traduites par l'identité. Cette non-observabilité provient de la remarque qu'on peut focaliser une infinité de fois sur tout dessein négatif.

Cette refocalisation possible sur chaque négatif évoque naturellement la logique linéaire polarisée [Lau02]. Cependant, ce n'est pas évident car la logique linéaire restreinte aux formules polarisées avec des décalages et des exponentielles a un statut raisonnable en dehors de la logique linéaire polarisée. Dans ce cadre, les règles exponentielles seraient remplacées par :

$$\begin{array}{ll}
 \text{promotion} & \frac{\Gamma, P \vdash \Delta;}{\Gamma \vdash \Delta; !P^\perp} \qquad \text{dérélliction} \quad \frac{\Gamma \vdash \Delta; P}{\Gamma, !P^\perp \vdash \Delta;} \\
 \text{affaiblissement} & \frac{\Gamma \vdash \Delta;}{\Gamma, P \vdash \Delta;} \qquad \text{contraction} \quad \frac{\Gamma, P, P \vdash \Delta;}{\Gamma, P \vdash \Delta;}
 \end{array}$$

Dans ce modèle, les exponentielles sont égales aux décalages qu'il faut donc enlever.

Les différences entre ces interprétations (LL ou LLP) devraient être détectables dans les définitions d'uniformité.

Pour finir cette rapide section sur l'interprétation des exponentielles, on peut remarquer que l'isomorphisme de la logique linéaire :

$$!(A \& B) = !A \otimes !B$$

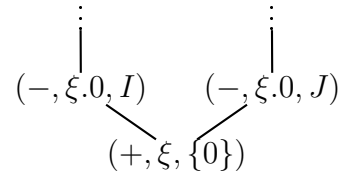
se reflète dans l'isomorphisme linéaire

$$\downarrow(A \& B) = \downarrow A \otimes \downarrow B$$

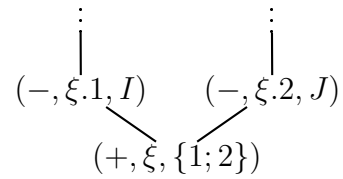
pour des comportements négatifs A et B puisque, sur les comportements négatifs,

$$!G = \downarrow G$$

En ludique, cet isomorphisme n'est pas une égalité puisque les desseins semi-simples de $\downarrow(A \& B)$ sont de la forme



alors que ceux de $\downarrow A \otimes \downarrow B$ sont de la forme :



L'isomorphisme est reflété dans le fait qu'un dessein incarné de $(\downarrow(A \& B))^{\perp}$ dans lequel on change

- l'action $(-, \xi, \{0\})$ en $(-, \xi, \{1; 2\})$
 - les actions $(+, \xi.0.*, I)$ en $(+, \xi.1.*, I)$ si I appartient au répertoire de A
 - les actions $(+, \xi.0.*, J)$ en $(+, \xi.1.*, J)$ si J appartient au répertoire de B
- est un dessein incarné de $(\downarrow A \otimes \downarrow B)^{\perp}$ et réciproquement.

Conclusion

Cette thèse apporte plusieurs contributions à la ludique ou plus généralement à la sémantique de jeux de la logique linéaire :

1. Un modèle probabiliste avec une notion de coefficients originale apportant, via une géométrie particulière des coefficients, de nouvelles techniques de preuves et permettant d'étendre des résultats difficilement généralisables.
2. Un théorème de complétude pour la partie linéaire vis-à-vis de la logique $MALL_2$ qui montre une certaine conservativité du modèle probabiliste par rapport au modèle simple de départ.
3. Une première interprétation des exponentielles dans un cadre interactif vérifiant les théorèmes analytiques.

Quelques développements futurs possibles apparaissent :

1. La question la plus importante est naturellement d'énoncer et de montrer un théorème de complétude pour la partie exponentielle. Il faudrait vérifier si l'interprétation des décalages est suffisamment différente de celle des exponentielles pour accommoder toute la logique linéaire avec décalages et exponentielles ou s'il faut restreindre les formules aux formules polarisées de la logique linéaire sans décalages.
2. Ce théorème de complétude doit s'énoncer par rapport à une certaine définition des exponentielles. Il est possible que la restriction à certaines classes de desseins vérifiant les théorèmes analytiques et closes par réduction permettent de définir différentes exponentielles. Une possibilité est la classe des desseins tels que le nombre de répétitions d'un lieu donné dans chaque branche est borné.
3. Une autre question naturelle est celle de la surjectivité de la fonction *Proba* (remarque 5.3 page 135) qui associe un comportement probabiliste à un comportement simple. Elle semble plausible mais les tentatives pour la prouver sont restées infructueuses.

Bibliographie

- [AHM98] Samson ABRAMSKY, Kohei HONDA, et Guy MCCUSKER. A fully abstract game semantics for general references. Dans *Proceedings, Thirteenth Annual IEEE Symposium on Logic in Computer Science*, 1998.
- [AJ94] Samson ABRAMSKY et Radha JAGADEESAN. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59(2) :543–574, 1994.
- [AJM00] Samson ABRAMSKY, Radha JAGADEESAN, et Pasquale MALACARIA. Full abstraction for PCF. *Information and Computation*, 163 :409–470, 2000.
- [AM99] Samson ABRAMSKY et Paul-André MELLIÈS. Concurrent games and full completeness. Dans *Logic in Computer Science*, 1999. Trente.
- [And90] Jean-Marc ANDREOLI. *Proposition pour une synthèse des paradigmes de la programmation logique et de la programmation par objets*. Thèse de doctorat, Université Paris VI, 1990.
- [BC82] Gérard BERRY et Pierre-Louis CURIEN. Sequential algorithms on concrete data structures. *Theoretical Computer Science*, 20 :265–321, 1982.
- [BDER97a] Patrick BAILLOT, Vincent DANOS, Thomas EHRHARD, et Laurent REGNIER. Believe it or not, ajm’s games models is a model of classical linear logic. Dans *Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 68–75. IEEE Computer Society Press, 1997.
- [BDER97b] Patrick BAILLOT, Vincent DANOS, Thomas EHRHARD, et Laurent REGNIER. Timeless games. Dans *Computer Science Logic*, 1997.
- [BE99] Nuno BARREIRO et Thomas EHRHARD. Quantitative semantics revisited. Dans *Typed Lambda-Calculi and Applications*, pages 40–53. Springer Verlag, 1999.
- [Ber79] Gérard BERRY. *Modèles complètement adéquats et stables des λ -calculs typés*. Thèse de doctorat d’état, Université de Paris 7, 1979.
- [Bla92] Andreas BLASS. A game semantics for linear logic. *Annals of Pure and Applied Logic*, 56 :183–220, 1992. Special Volume dedicated to the memory of John Myhill.
- [CH98] Pierre-Louis CURIEN et Hugo HERBELIN. Computing with abstract Böhm trees. Dans *Third Fuji International Symposium on Functional and Logic Programming (Kyoto, 1998)*, pages 20–39. World Sci. Publishing, River Edge, NJ, 1998.

- [CPW00] Pierre-Louis CURIEN, Gordon PLOTKIN, et Glynn WINSKEL. Bistructures, bidomains and linear logic. Dans *Proof, Language, and Interaction : Essays in Honour of Robin Milner*. MIT press, 2000.
- [Cur98] Pierre-Louis CURIEN. Abstract Böhm trees. *Math. Structures Comput. Sci.*, 8(6) :559–591, 1998.
- [Cur04a] Pierre-Louis CURIEN. Introduction to linear logic and ludics, part i. manuscript, 2004.
- [Cur04b] Pierre-Louis CURIEN. Introduction to linear logic and ludics, part ii. manuscript, 2004.
- [Dan90a] Vincent DANOS. Lambda-calculus and its dynamic algebra I : basic results. Preprint 7, Équipe de Logique Mathématique, University of Paris VII, février 1990.
- [Dan90b] Vincent DANOS. *La Logique Linéaire Appliquée à l'étude de Divers Processus de Normalisation (Principalement du λ -Calcul)*. Thèse de doctorat, University of Paris VII, juin 1990.
- [DH02] Vincent DANOS et Russell HARMER. Probabilistic games semantics. *ACM Transactions on Computational Logic*, 2002.
- [DHR96] Vincent DANOS, Hugo HERBELIN, et Laurent REGNIER. Games semantics and abstract machines. Dans *Logic in Computer Science*, 1996.
- [DJS93] Vincent DANOS, Jean-Baptiste JOINET, et Harold SCHELLINX. The structure of exponentials : Uncovering the dynamics of linear logic proofs. Dans G. GOTTLÖB, A. LEITSCH, et D. MUNDICI, éditeurs, *Proceedings of the Third Kurt Gödel Colloquium on Computational Logic and Proof Theory*, pages 159–171, Brno, Czech Republic, août 1993. Springer-Verlag LNCS 348.
- [DR95] Vincent DANOS et Laurent REGNIER. Proof-nets and Hilbert space. Dans J.-Y. GIRARD, Y. LAFONT, et L. REGNIER, éditeurs, *Advances in Linear Logic*, pages 307–328. Cambridge University Press, 1995. Proceedings of the Workshop on Linear Logic, Ithaca, New York, June 1993.
- [Ehr95] T. EHRHARD. Hypercoherences : A strongly stable model of linear logic. Dans J.-Y. GIRARD, Y. LAFONT, et L. REGNIER, éditeurs, *Advances in Linear Logic*, pages 83–108. Cambridge University Press, 1995. Proceedings of the Workshop on Linear Logic, Ithaca, New York, June 1993.
- [Ehr01] Thomas EHRHARD. On koethe sequence spaces and linear logic. *Mathematical Structures in Computer Science*, 12 :579–623, 2001. Cambridge University Press.
- [Ehr04] Thomas EHRHARD. Finiteness spaces. *Mathematical Structures in Computer Science*, 2004. Cambridge University Press. To appear.
- [ER03] Thomas EHRHARD et Laurent REGNIER. The differential lambda calculus. *Theoretical Computer Science*, 309 :1–41, 2003.
- [Esc98] Martin H. ESCARDÓ. A metric model of PCF. Laboratory for Foundations of Computer Science, University of Edinburgh. Unpublished research note, presented at the Workshop on Realizability Semantics and Applications, 1999, April 1998.

- [Fag02] Claudia FAGGIAN. *On the dynamics of ludics, a study of interaction*. Thèse de doctorat, Institut de Mathématiques de Luminy, 2002.
- [FFDQ02] Claudia FAGGIAN, Marie-Renée FLEURY-DONNADIEU, et Myriam QUATRINI. An introduction to uniformity in ludics. Dans Thomas EHRHARD, Jean-Yves GIRARD, Paul RUET, et Phil SCOTT, éditeurs, *Linear Logic in Computer Science*. 2002.
- [FH02] Claudia FAGGIAN et Martin HYLAND. Designs, disputes and strategies. Dans *Computer Science Logic*, 2002.
- [FQ04] Marie-Renée FLEURY et Myriam QUATRINI. First order in ludics. *Mathematical Structures in Computer Science*, 14 :189–213, 2004. Cambridge University Press.
- [Gir87] Jean-Yves GIRARD. Linear logic. *Theoretical Computer Science*, 50 :1–102, 1987.
- [Gir89a] Jean-Yves GIRARD. Geometry of interaction I : Interpretation of system F. Dans C. BONOTTO, R. FERRO, S. VALENTINI, et A. ZANARDO, éditeurs, *Logic Colloquium '88*, pages 221–260. North-Holland, 1989.
- [Gir89b] Jean-Yves GIRARD. Towards a geometry of interaction. Dans J. W. GRAY et A. SCEDROV, éditeurs, *Categories in Computer Science and Logic*, pages 69–108. American Mathematical Society, 1989. Proceedings of the AMS-IMS-SIAM Joint Summer Research Conference, June 14–20, 1987, Boulder, Colorado ; Contemporary Mathematics Volume 92.
- [Gir90] Jean-Yves GIRARD. Geometry of interaction II : Deadlock-free algorithms. Dans P. MARTIN-LÖF et G. MINTS, éditeurs, *COLOG-88*, pages 76–93. Springer-Verlag LNCS 417, 1990.
- [Gir91a] Jean-Yves GIRARD. A new constructive logic : classical logic. *Mathematical Structures in Computer Science*, 1(3) :255–296, 1991.
- [Gir91b] Jean-Yves GIRARD. Quantifiers in linear logic II. Dans G. CORSI et G. SAMBIN, éditeurs, *Nuovi problemi della logica e della filosofia della scienza*, volume II, Bologna, Italy, 1991. CLUEB.
- [Gir95a] Jean-Yves GIRARD. Geometry of interaction III : The general case. Dans J.-Y. GIRARD, Y. LAFONT, et L. REGNIER, éditeurs, *Advances in Linear Logic*, pages 329–389. Cambridge University Press, 1995. Proceedings of the Workshop on Linear Logic, Ithaca, New York, June 1993.
- [Gir95b] Jean-Yves GIRARD. Linear logic : Its syntax and semantics. Dans J.-Y. GIRARD, Y. LAFONT, et L. REGNIER, éditeurs, *Advances in Linear Logic*, pages 1–42. Cambridge University Press, 1995. Proceedings of the Workshop on Linear Logic, Ithaca, New York, June 1993.
- [Gir95c] Jean-Yves GIRARD. On denotational completeness. Manuscript, 1995.
- [Gir96a] Jean-Yves GIRARD. Coherent banach spaces : a continuous denotational semantics. unpublished manuscript, 1996.
- [Gir96b] Jean-Yves GIRARD. Constructivité : vers une dualité moniste. unpublished manuscript, 1996.

- [Gir96c] Jean-Yves GIRARD. Proof-nets : the parallel syntax for proof-theory. Dans *Logic and algebra*, pages 97–124. Dekker, New York, 1996.
- [Gir98] Jean-Yves GIRARD. On the meaning of logical rules II : multiplicatives and additives. unpublished manuscript, 1998.
- [Gir99] Jean-Yves GIRARD. On the meaning of logical rules I : syntax vs. semantics. Dans BERGER et SCHWICHTENBERG, éditeurs, *Computational Logic*, 1999.
- [Gir01] Jean-Yves GIRARD. Locus solum : from the rules of logic to the logic of rules. *Math. Structures Comput. Sci.*, 11(3) :301–506, 2001.
- [GLT88] Jean-Yves GIRARD, Yves LAFONT, et Paul TAYLOR. *Proofs and Types*. Cambridge Tracts in Theoretical Computer Science 7. Cambridge University Press, 1988.
- [Har99] Russel HARMER. *Games and full abstraction for nondeterministic languages*. Thèse de doctorat, University of London, 1999.
- [HO00] J. Martin E. HYLAND et C.-H. Luke ONG. On full abstraction for PCF : I, II, and III. *Information and Computation*, 163(2) :285–408, décembre 2000.
- [Joi93] J.-B. JOINET. *Étude de la Normalisation du Calcul des Séquents Classique à Travers la Logique Linéaire*. Thèse de doctorat, University of Paris VII, 1993.
- [JP89] Cliff B. JONES et Gordon W. PLOTKIN. A probabilistic powerdomain of evaluations. Dans *Fourth Annual Symposium on Logic in Computer Science*, pages 189–195. IEEE Computer Society Press, 1989.
- [Lau02] Olivier LAURENT. *Etude de la polarisation en logique*. Thèse de doctorat, Université Aix-Marseille II, mars 2002.
- [Lau04] Olivier LAURENT. Polarized games. *Annals of Pure and Applied Logic*, 2004. To appear.
- [LMS95] Patrick LINCOLN, John MITCHELL, et Andre SCEDROV. Stochastic interaction and linear logic. Dans J.-Y. GIRARD, Y. LAFONT, et L. REGNIER, éditeurs, *Advances in Linear Logic*, pages 147–166. Cambridge University Press, 1995. Proceedings of the Workshop on Linear Logic, Ithaca, New York, June 1993.
- [Lon01] John LONGLEY. Matching typed and untyped realizability. Dans Lars BIRKEDAL, Jaap van OOSTEN, Giuseppe ROSOLINI, et Dana S. SCOTT, éditeurs, *Electronic Notes in Theoretical Computer Science*, volume 23. Elsevier, 2001.
- [LTdF01] Olivier LAURENT et Lorenzo Tortora de FALCO. Slicing polarized additive normalization. Quaderno 12, Istituto per le Applicazioni del Calcolo (CNR), Roma, Italia, mars 2001. A paraître dans *Linear Logic in Computer Science* (London Mathematical Society Lecture Notes Series - CUP).
- [Mac71] Saunders MACLANE. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971.
- [Mil77] Robin MILNER. Fully abstract models of typed λ -calculi. *Theoretical Computer Science*, 4 :1–22, 1977.

- [Plo77] Gordon W. PLOTKIN. LCF considered as a programming language. *Theoretical Computer Science*, 5, 1977.
- [Plo80] Gordon W. PLOTKIN. Lambda-definability in the full type hierarchy. Dans *To H.B. Curry : Essays on combinatory logic, lambda calculus and formalism*, pages 363–373. Academic Press, 1980.
- [Reg92] Laurent REGNIER. *λ -Calcul et Réseaux*. Thèse de doctorat, University of Paris VII, 1992.
- [Sie92] Karl SIEBER. Reasoning about sequential functions via logical relations. Dans *Applications of categories in computer science*, pages 258–269. Cambridge University Press, 1992.

Index

A

action 27, 55
 avec pointeurs 174
 simple 69
 visible 32, 81, 176, 210
 adjonctions 42
 approximations d'un coefficient 71
 arbre
 avec pointeurs 174
 co-fini 69
 de Böhm abstrait 186
 exponentiel 208
 probabiliste 75
 simple 29, 55

B

base 27
 interne 219
 réseau 32, 81, 176, 210
 unaire 27
 vide 27
 bi-incarnation 46, 156
 bi-comportement 45, 156, 235
 s-régulier 161
 bi-éthique 45, 61, 153, 234
 complète 157
 bi-orthogonal
 linéaire 195, 228
 probabiliste 162
 borne inférieure 159

C

c-chronique 97, 209
c-préfixe 98
c-tranche 97, 209
 chemin 127
 choix 115
 incompatible 115
 chronique

avec pointeurs 175
 simple 28, 56
 co-finitude 69
 coefficient 70
 fini 70
 négatif 75
 positif 75
 quasi-simple 115
 régulier 70
 sous-jacent 152, 234
 cohérence
c-chroniques 98
c-tranches 101
 combinaison linéaire 79, 131, 208
 comportement 37, 130
 connexe 40, 144, 198, 229
 disjoint 40, 144, 198, 229
 étranger 43, 198, 230
 linéaire 196, 227
 multiple 195, 227
 s-régulier 135
 connecteur
avec 41, 159, 202, 232, 237
bémol 201, 231
décalage 39, 59, 157, 199, 231, 236
dièse 201, 231
 exponentielles 236
intersection . 40, 46, 158, 201, 232,
237
nœud papillon 46, 159, 238
plus 41, 159, 202, 232, 237
 s-régulier 136, 161
tenseur 42, 58, 147, 203, 233
union ... 41, 46, 159, 202, 232, 237
 coupure 30
 couverture 116
 développée 123
c-tranche

- d'un dessein 97, 209
- D**
- décalage 39, 145, 199, 231
- décomposition additive 41
- délocalisation 39
- démon 27
- focalisé 55
- dessein 81
- avec pointeurs 176
- complet 80
- continu 127
- couvert 116
- developpé 118
- exact 48, 166
- exponentiel 209
- gagnant 48, 166
- immédiatement total 81, 208
- incarné 38, 196
- linéaire 195, 227
- matériel 38, 196
- obstiné 48, 166
- partiel 44, 152, 233
- positivement probabiliste ... 82, 210
- présent 59, 166
- principal 32
- probabiliste 77
- propre 46
- quasi-simple 115
- semi-simple 79
- simple 30, 56
- uniforme 45, 156, 235
- universel 106, 222
- desseins
- compatibles 193
- orthogonaux 96, 183
- pleinement compatibles ... 109, 225
- dessin 27
- développement 120
- distance
- pré-distance 152
- sur les coefficients 150
- sur les desseins 151
- E**
- ensemble
- des scalaires 103, 218
- équation
- générique 95
- évaluation 96
- équivalence
- partielle 44
- éthique 37, 61, 130, 195, 227
- complète 38, 134, 197, 228
- linéaire complète 197, 228
- partielle 44
- exponentielles 200, 231
- F**
- fax
- plus normalisant 126
- formule
- de la logique linéaire 20
- Π_1 51
- H**
- historique 178
- arbres de Böhm abstraits 186
- I**
- incarnation 37, 133, 196, 228
- d'un comportement 38, 196
- partiel 45
- inclusion 30, 76, 193, 209
- pleine 101
- indépendance 43, 145, 198, 230
- intersection 36, 193
- pleine 110, 226
- J**
- justification 28, 56, 175
- L**
- lieu 27
- lieux disjoints 27
- visible 32, 81, 176, 210
- ludique
- avec pointeurs 190
- probabiliste 65
- simple 19
- M**
- MAAL₂ 51

- machine abstraite 180
 MALL₂ 50
- N**
- négation 21
 normalisation 34, 57
 de la ludique avec pointeurs 183
 de la ludique exponentielle 214
 de la ludique probabiliste 91
 de tranches 33
 d'un réseau clos 33
 par codage 82
 par c -tranches 99
 par quotient d'ordres 33
 par tranches 97
- O**
- obstination 48, 166
 opposant universel 107, 223
 ordre
 extensionnel .. 35, 60, 103, 191, 216
 stable 34, 60, 102, 190, 216
 sur les couvertures 116
 orthogonal
 d'un dessein 35
 d'une bi-éthique 45, 154, 234
 d'une éthique 37, 130, 195, 227
 orthogonalité 33, 59
 oubli d'un historique 183
 oubli simple
 d'un dessein quasi-simple 115
- P**
- parcimonie 48, 166
 partie semi-simple 131
 plongement
 canonique
 des dessins avec pointeurs 215
 des dessins simples 95
 plus normalisant 126
 poids
 d'un coefficient 71
 d'un dessein quasi-simple 115
 d'une branche 73
 pointeur 173, 190
 associé à une base interne 220
- position 186
 pré-chronique
 avec pointeurs 175
 simple 28, 56
 pré-réseau 30, 81, 176, 210
 principe
 des équations génériques
 ludique avec pointeurs 189
 ludique exponentielle 215
 ludique probabiliste 96
 projection
 sans coefficients 209
 semi-simple 131
 simple 77
 complète 137
- R**
- réseau
 clos 32
 de réseaux 36
 ramification 27
 règle réversible 21
 répertoire 39
 d'un comportement ... 40, 143, 197,
 229
 répertoires indépendants ... 43, 145,
 198, 230
 négatif 28
 réseau 32, 176
 convergent 57
 réservoir 43
 résidu 122
- S**
- saut 179
 scalaire 69
 maximal 223
 séquent
 de bi-comportements 47, 160
 de comportements 47
 superposition 135
- T**
- théorèmes analytiques 35, 103, 191, 216
 traduction
 positivement probabiliste 94

vers la ludique avec pointeurs...	212
inverse.....	213
vers la ludique simple.....	88
inverse.....	90
vers les arbres de Böhm abstraits	188
inverse.....	189
tranche.....	29, 56
d'un arbre simple.....	30
exacte.....	48, 166
linéaire.....	30, 56
tranchette.....	51
U	
uniformité.....	45, 156, 235
union	
de c -tranches.....	101
de desseins.....	34
V	
vue.....	187

Liste des symboles

L	ordre ext.....	35	$\bigcup^{\perp\perp}$	<i>union</i>	41, 202, 232
$\mathbf{0}$	feuille nulle.....	70	$\mathcal{U}\mathcal{D}_{\mathcal{D}}$	opposant universel.	107, 223
$\mathbf{1}$	feuille vide.....	70	$[x \leftarrow \phi]$...	renommage.....	186
$\alpha_{\mathcal{D}}$	scalaire séparateur.	107, 223			
Chron_{β}	chroniques.....	28, 56, 175			
$\text{Cl}(E)$	combinaisons linéaires...	131			
$\text{Coef}_X(\mathcal{D})$.	coeff. sous-jacent ..	152, 234			
$\mathcal{D}\text{ai}^-$	démon négatif.....	143, 229			
$\mathcal{D}\text{ai}$	le démon.....	31			
$\mathcal{D}_{\mathcal{C}}$	dessein couvert.....	116			
\mathfrak{F}_{ξ}	démon focalisé.....	55			
$ \mathcal{D} _{\mathbb{G}}$	incarnation....	37, 133, 196			
$\mathcal{D}_s(\xi_c)$	codage d'un scalaire	86, 211			
$E^{\perp p}$	orthogonal probabiliste .	134			
$E^{\perp s}$	orthogonal simple.....	134			
$\mathcal{ET}(\mathcal{D})$	tranches de \mathcal{D}	51			
$\mathcal{ET}_{\downarrow}(\mathcal{D})$...	tranchettes de \mathcal{D}	51			
$\mathcal{ET}_{\downarrow}^{pr}(\mathcal{D})$..	tranchettes propres de \mathcal{D} .	51			
$\mathfrak{F}(\mathcal{D})$	adjoint du <i>tenseur</i>	43			
$\mathfrak{F}\text{id}^-$	divergence négative.....	55			
$\mathfrak{F}\text{id}^-$	divergence négative.....	75			
$\mathfrak{F}\text{id}$	divergence.....	29, 75			
\preccurlyeq	ordre ext.	60, 103, 191, 216			
\sqsubseteq	ordre st...	60, 102, 190, 216			
\cap	<i>intersection</i> ...	40, 201, 232			
$\text{Lin}_{\mathbb{G}}$	desseins linéaires ..	195, 227			
\bowtie	un \mathfrak{A} locatif ...	42, 203, 233			
$\text{Os}(\mathcal{D})$	oubli simple.....	115			
$\mathcal{P}_{fin}(X)$...	parties finies de X ..	40, 197			
Point.....	mise des pointeurs.....	190			
Proba(\mathbb{G}) .	desseins probabilistes ...	135			
Scal(\mathcal{D})...	scalaires de \mathcal{D}	103, 218			
Simp(\mathbb{G}) ..	desseins simples.....	135			
$\mathfrak{S}\mathfrak{k}\mathfrak{u}\mathfrak{n}\mathfrak{k}$	sconse.....	30			
$\text{Sp}(E)$	superposition de desseins	135			
\otimes	<i>tenseur</i>	43			
$\mathcal{U}\mathcal{D}_{\beta}(\alpha)$...	dessein universel ...	106, 222			

Résumé

La ludique, introduite par Jean-Yves Girard, est un modèle de la logique linéaire sans exponentielles. En vue de la modélisation des exponentielles, cette thèse propose deux extensions successives de la ludique. La première extension étend la ludique en un modèle probabiliste original conservant la plupart des théorèmes du modèle initial. La seconde extension reprend ce modèle probabiliste et utilise des pointeurs, introduits dans les jeux de Hyland et Ong, ce qui permet de considérer les répétitions et donc de modéliser les exponentielles.

Un résultat de complétude est montré pour la ludique probabiliste vis-à-vis de la ludique et de la logique $MALL_2$. La ludique exponentielle proposée vérifie les théorèmes principaux de la ludique : les théorèmes analytiques. En particulier, ce modèle montre une interprétation de la logique linéaire avec des répétitions gardant une topologie séparée. De plus, les coefficients introduits apporte à la ludique une “plus grande séparation” que dans le modèle d’origine.

Abstract

Ludics, introduced by Jean-Yves Girard, is a model of linear logic without exponentials. In the purpose of modeling exponentials, this thesis proposes two successive extensions of ludics. The first extension extends ludics in an original probabilistic model preserving the majority of the theorems of the initial model. The second extension takes again this probabilistic model and uses pointers, introduced in Hyland and Ong’s games, which makes it possible to consider repetitions and thus to model exponentials.

A completeness result is shown for probabilistic ludics with respect to ludics and to $MALL_2$. The proposed exponential ludics satisfies the main theorems of ludics : the analytical theorems. In particular, this model exhibits an interpretation of linear logic with repetitions while keeping a separated topology. Moreover, the introduced coefficients yield a “greater separation” than in the original model.