Sémantique des jeux

(notes de cours du DEA de Programmation)

Olivier LAURENT
Preuves Programmes Systèmes
`Olivier.Laurent@pps.jussieu.fr`

Version 2004

La sémantique des jeux a permis la modélisation de différentes primitives de programmation : fonctions [1, 9], contrôle [11], états [3], non-déterminisme [7, 8], probabilités [5], ... Nous ne nous intéresserons ici qu'aux deux premiers qui forment le noyau de la théorie.

En plus de sa souplesse et de son adaptabilité, la sémantique des jeux a connu son succès parce qu'elle permet d'obtenir des modèles précis et proches des langages (résultats de complétude).

Pour d'autres références sur la sémantique des jeux, on pourra consulter :
– l'article originel sur le modèle HO [9]
– l'article originel sur le modèle AJM [1]
– la thèse de Guy McCusker [16]
– la thèse de Jim Laird [12]
– la thèse de Russ Harmer [7]
– les notes de cours de Samson Abramsky et Guy McCusker [2]
– les transparents de cours de jeux de Vincent Danos [4]

1 Modèle de jeux HO sans réponses du λ-calcul

1.1 Exemples, idées, intuitions

Contrairement aux modèles ensemblistes (et à ceux de théorie des domaines), les modèles de jeux ne sont pas des modèles de *fonctions* mais des modèles de *traces* : l'interprétation d'un programme est l'ensemble des traces d'exécution possibles de ce programme dans les différents environnements possibles.

1.1.1 Traces

Ces traces sont représentées dans un jeu à deux joueurs : Joueur (J) et Opposant (O), où Joueur (auquel on aura tendance à s'identifier) représente le programme et Opposant représente le contexte (ou l'environnement) d'évaluation.

On va considérer, pour les exemples, une extension du λ-calcul simplement typé avec des booléens : un type $\mathbb{B}$ avec deux constantes (valeurs) V et F et une instruction $\mathsf{if} \ldots \mathsf{then} \ldots \mathsf{else} \ldots$ dans un contexte d'appel par nom (il s'agit d'un sous-système fortement normalisant de PCF [18]).

Exemple 1 (Type $\mathbb{B}$)
Un programme de type $\mathbb{B}$ a soit la valeur V soit la valeur F. Une trace d'un tel programme est

représentée par :

$$\begin{array}{cc} & \mathbb{B} \\ \mathsf{O} & \mathsf{q} \\ \mathsf{J} & \mathsf{b} \end{array}$$

avec $\mathsf{b} = \mathsf{V}$ ou F selon le programme : l'environnement demande la valeur du programme (q) et le programme répond (b). Une autre trace possible est :

$$\begin{array}{cc} & \mathbb{B} \\ \mathsf{O} & \mathsf{q} \\ \mathsf{J} & \mathsf{b} \\ \mathsf{O} & \mathsf{q} \\ \mathsf{J} & \mathsf{b} \end{array}$$

si l'environnement redemande la valeur du programme.

Dans un cadre non déterministe, on pourrait avoir :

$$\begin{array}{cc} & \mathbb{B} \\ \mathsf{O} & \mathsf{q} \\ \mathsf{J} & \mathsf{V} \\ \mathsf{O} & \mathsf{q} \\ \mathsf{J} & \mathsf{V} \\ \mathsf{O} & \mathsf{q} \\ \mathsf{J} & \mathsf{F} \\ \mathsf{O} & \mathsf{q} \\ \mathsf{J} & \mathsf{V} \end{array}$$

Exemple 2 (Type $\mathbb{B} \times \mathbb{B}$)

Un programme de type $\mathbb{B} \times \mathbb{B}$ est donné par une paire de valeurs booléennes. Ce qui donne, dans le cas du couple (V, F) :

$$\begin{array}{cccc} & \mathbb{B} & \times & \mathbb{B} \\ \mathsf{O} & & & \mathsf{q} \\ \mathsf{J} & & & \mathsf{F} \\ \mathsf{O} & \mathsf{q} & & \\ \mathsf{J} & \mathsf{V} & & \end{array} \qquad \begin{array}{cccc} & \mathbb{B} & \times & \mathbb{B} \\ \mathsf{O} & \mathsf{q} & & \\ \mathsf{J} & \mathsf{V} & & \\ \mathsf{O} & & & \mathsf{q} \\ \mathsf{J} & & & \mathsf{F} \end{array} \qquad \begin{array}{cccc} & \mathbb{B} & \times & \mathbb{B} \\ \mathsf{O} & \mathsf{q} & & \\ \mathsf{J} & \mathsf{V} & & \\ \mathsf{O} & \mathsf{q} & & \\ \mathsf{J} & \mathsf{V} & & \\ \mathsf{O} & & & \mathsf{q} \\ \mathsf{J} & & & \mathsf{F} \end{array}$$

Que penser de la trace suivante ?

$$\begin{array}{cccc} & \mathbb{B} & \times & \mathbb{B} \\ \mathsf{O} & \mathsf{q} & & \\ \mathsf{J} & & & \mathsf{F} \end{array}$$

on va devoir imposer des contraintes sur la formation des traces. On ne veut pas que Joueur réponde la valeur du second booléen si on lui a demandé celle du premier...

Exemple 3 (Type $\mathbb{B} \to \mathbb{B}$)

Un programme de type $\mathbb{B} \to \mathbb{B}$ nécessite une valeur booléenne et rend un booléen calculé en fonction de cette entrée. La fonction *identité* comporte plusieurs traces selon la valeur de l'entrée.

Cette entrée dépend du contexte et est donc fournie par Opposant alors que la valeur finale est répondue par le programme (donc Joueur) :

$$\begin{array}{llcl}
 & \mathbb{B} & \to & \mathbb{B} \\
\mathsf{O} & & & \mathsf{q} \\
\mathsf{J} & \mathsf{q} & & \\
\mathsf{O} & \mathsf{V} & & \\
\mathsf{J} & & & \mathsf{V}
\end{array}
\qquad \text{et} \qquad
\begin{array}{llcl}
 & \mathbb{B} & \to & \mathbb{B} \\
\mathsf{O} & & & \mathsf{q} \\
\mathsf{J} & \mathsf{q} & & \\
\mathsf{O} & \mathsf{F} & & \\
\mathsf{J} & & & \mathsf{F}
\end{array}$$

Pour la négation, on obtient :

$$\begin{array}{llcl}
 & \mathbb{B} & \to & \mathbb{B} \\
\mathsf{O} & & & \mathsf{q} \\
\mathsf{J} & \mathsf{q} & & \\
\mathsf{O} & \mathsf{V} & & \\
\mathsf{J} & & & \mathsf{F}
\end{array}
\qquad \text{et} \qquad
\begin{array}{llcl}
 & \mathbb{B} & \to & \mathbb{B} \\
\mathsf{O} & & & \mathsf{q} \\
\mathsf{J} & \mathsf{q} & & \\
\mathsf{O} & \mathsf{F} & & \\
\mathsf{J} & & & \mathsf{V}
\end{array}$$

Si on s'intéresse à la fonction constante de valeur V, il existe deux "algorithmes" différents pour la calculer. Le premier regarde la valeur de l'argument puis rend la valeur V :

$$\begin{array}{llcl}
 & \mathbb{B} & \to & \mathbb{B} \\
\mathsf{O} & & & \mathsf{q} \\
\mathsf{J} & \mathsf{q} & & \\
\mathsf{O} & \mathsf{V} & & \\
\mathsf{J} & & & \mathsf{V}
\end{array}
\qquad \text{et} \qquad
\begin{array}{llcl}
 & \mathbb{B} & \to & \mathbb{B} \\
\mathsf{O} & & & \mathsf{q} \\
\mathsf{J} & \mathsf{q} & & \\
\mathsf{O} & \mathsf{F} & & \\
\mathsf{J} & & & \mathsf{V}
\end{array}$$

Mais un autre programme possible donne la trace suivante :

$$\begin{array}{llcl}
 & \mathbb{B} & \to & \mathbb{B} \\
\mathsf{O} & & & \mathsf{q} \\
\mathsf{J} & & & \mathsf{V}
\end{array}$$

La réponse est donnée sans même regarder l'argument. Cette distinction existe dans le langage, entre $\lambda x.\mathsf{if}\ x\ \mathsf{then}\ \mathsf{V}\ \mathsf{else}\ \mathsf{V}$ (constante "stricte") et $\lambda x.\mathsf{V}$ (constante "par vocation").

Un programme peut également regarder deux fois la valeur de son argument avant de répondre :

$$\begin{array}{llcl}
 & \mathbb{B} & \to & \mathbb{B} \\
\mathsf{O} & & & \mathsf{q} \\
\mathsf{J} & \mathsf{q} & & \\
\mathsf{O} & \mathsf{V} & & \\
\mathsf{J} & \mathsf{q} & & \\
\mathsf{O} & \mathsf{V} & & \\
\mathsf{J} & & & \mathsf{F}
\end{array}
\qquad \text{et} \qquad
\begin{array}{llcl}
 & \mathbb{B} & \to & \mathbb{B} \\
\mathsf{O} & & & \mathsf{q} \\
\mathsf{J} & \mathsf{q} & & \\
\mathsf{O} & \mathsf{F} & & \\
\mathsf{J} & \mathsf{q} & & \\
\mathsf{O} & \mathsf{F} & & \\
\mathsf{J} & & & \mathsf{V}
\end{array}$$

Il y a bien d'autres programmes possibles avec ce type, par exemple :

	$\mathbb{B}$	$\to$	$\mathbb{B}$
O			q
J	q		
O	V		
J			V

	$\mathbb{B}$	$\to$	$\mathbb{B}$
O			q
J	q		
O	F		
J	q		
O	V		
J			V

	$\mathbb{B}$	$\to$	$\mathbb{B}$
O			q
J	q		
O	F		
J	q		
O	F		
J	q		
O	V		
J			V

$\dots$

Exercice 1 (Traces et termes)
Donner des termes qui engendrent les différents ensembles de traces donnés précédemment. Est-ce possible en λ-calcul avec booléens ? dans un autre langage ?

Exemple 4 (Type $\mathbb{B} \to \mathbb{B} \times \mathbb{B}$)
Un programme de type $\mathbb{B} \to \mathbb{B} \times \mathbb{B}$ calcule deux valeurs booléennes, fournit celle qu'Opposant demande (comme dans le cas du type $\mathbb{B} \times \mathbb{B}$) mais fait appel à une entrée de type $\mathbb{B}$ (comme pour $\mathbb{B} \to \mathbb{B}$) :

	$\mathbb{B}$	$\to$	$\mathbb{B}$	$\times$	$\mathbb{B}$
O			q		
J	q				
O	V				
J			V		

	$\mathbb{B}$	$\to$	$\mathbb{B}$	$\times$	$\mathbb{B}$
O					q
J	q				
O	V				
J					F

	$\mathbb{B}$	$\to$	$\mathbb{B}$	$\times$	$\mathbb{B}$
O			q		
J	q				
O	V				
J			V		
O					q
J					F

Si on cherche le terme correspondant, les deux premières traces suggèrent $\lambda x.\mathsf{if}\ x\ \mathsf{then}\ (\mathsf{V}, \mathsf{F})\ \mathsf{else}\ \dots$, mais ce terme engendre-t-il la troisième trace ?

Exemple 5 (Type $\mathbb{B} \to \mathbb{B} \to \mathbb{B}$)
Un programme de type $\mathbb{B} \to \mathbb{B} \to \mathbb{B}$ (ou $\mathbb{B} \times \mathbb{B} \to \mathbb{B}$ par décurryfication) utilise deux entrées booléennes pour calculer son résultat. C'est typiquement le cas d'un programme pour calculer le "et" de deux booléens. On en trouve différentes variantes parmi lesquelles :

- le "et gauche" (strict)

	$\mathbb{B}$	$\to$	$\mathbb{B}$	$\to$	$\mathbb{B}$
O					q
J	q				
O	V				
J			q		
O			V		
J					V

	$\mathbb{B}$	$\to$	$\mathbb{B}$	$\to$	$\mathbb{B}$
O					q
J	q				
O	F				
J			q		
O			V		
J					F

	$\mathbb{B}$	$\to$	$\mathbb{B}$	$\to$	$\mathbb{B}$
O					q
J	q				
O	F				
J			q		
O			F		
J					F

– le “et droit” (strict)

	$\mathbb{B}$ $\to$	$\mathbb{B}$ $\to$	$\mathbb{B}$
O			q
J		q	
O		V	
J	q		
O	V		
J			V

	$\mathbb{B}$ $\to$	$\mathbb{B}$ $\to$	$\mathbb{B}$
O			q
J		q	
O		V	
J	q		
O	F		
J			F

	$\mathbb{B}$ $\to$	$\mathbb{B}$ $\to$	$\mathbb{B}$
O			q
J		q	
O		F	
J	q		
O	F		
J			F

– le “et gauche” (non strict)

	$\mathbb{B}$ $\to$	$\mathbb{B}$ $\to$	$\mathbb{B}$
O			q
J	q		
O	V		
J		q	
O		V	
J			V

	$\mathbb{B}$ $\to$	$\mathbb{B}$ $\to$	$\mathbb{B}$
O			q
J	q		
O	F		
J			F

	$\mathbb{B}$ $\to$	$\mathbb{B}$ $\to$	$\mathbb{B}$
O			q
J	q		
O	V		
J		q	
O		F	
J			F

Il n’y a pas de différences majeures avec le type $\mathbb{B} \to \mathbb{B}$.

Exemple 6 (Type $(\mathbb{B} \to \mathbb{B}) \to \mathbb{B}$)
Un programme de type $(\mathbb{B} \to \mathbb{B}) \to \mathbb{B}$ est une fonctionnelle, les choses se compliquent... Parmi les plus simples, on trouve la fonctionnelle qui passe une valeur fixée (V par exemple) à la fonction qu’elle reçoit en argument et rend le résultat obtenu :

	($\mathbb{B}$ $\to$	$\mathbb{B}$) $\to$	$\mathbb{B}$
O			q
J		q	
O	q		
J	V		
O		V	
J			V

	($\mathbb{B}$ $\to$	$\mathbb{B}$) $\to$	$\mathbb{B}$
O			q
J		q	
O	q		
J	V		
O		F	
J			F

	($\mathbb{B}$ $\to$	$\mathbb{B}$) $\to$	$\mathbb{B}$
O			q
J		q	
O		F	
J			F

Cela correspond au terme $\lambda f.(f)$V qui est une variante de l’entier de Church 1. On peut faire de même pour la variante de l’entier 2 qu’est $\lambda f.(f)(f)$V :

	($\mathbb{B}$ $\to$	$\mathbb{B}$) $\to$	$\mathbb{B}$
O			q
J		q	
O	q		
J		q	
O	q		
J	V		
O		V	
J	V		
O		V	
J			V

	($\mathbb{B}$ $\to$	$\mathbb{B}$) $\to$	$\mathbb{B}$
O			q
J		q	
O	q		
J		q	
O	q		
J	V		
O		F	
J	F		
O		V	
J			V

	($\mathbb{B}$ $\to$	$\mathbb{B}$) $\to$	$\mathbb{B}$
O			q
J		q	
O	q		
J		q	
O	q		
J	V		
O		F	
J	F		
O		F	
J			F

Un autre programme intéressant à ce type est catch qui regarde si la fonction qu'il reçoit en argument est stricte (*i.e.* utilise elle-même son argument) ou pas :

	($\mathbb{B}$	$\to \mathbb{B}$)	$\to \mathbb{B}$
O			q
J		q	
O		b	
J			F

et

	($\mathbb{B}$	$\to \mathbb{B}$)	$\to \mathbb{B}$
O			q
J		q	
O	q		
J			V

Exemple 7 (Type $(\mathbb{B} \to \mathbb{B} \to \mathbb{B}) \to \mathbb{B}$)
Ce type ne nous intéresse que pour un exemple particulier : le *goûteur de "et"* :

	($\mathbb{B}$	$\to \mathbb{B}$	$\to \mathbb{B}$)	$\to \mathbb{B}$
O				q
J			q	
O	q			
J	b_1			
O		q		
J		b_2		
O			b_3	
J				V

et

	($\mathbb{B}$	$\to \mathbb{B}$	$\to \mathbb{B}$)	$\to \mathbb{B}$
O				q
J			q	
O		q		
J		b_1		
O	q			
J	b_2			
O			b_3	
J				F

Intuitivement, si on lui donne comme argument un programme pour le "et gauche" il répondra V et si on lui donne un "et droit" il répondra F.

Exemple 8 (Type $((\mathbb{B} \to \mathbb{B}) \to \mathbb{B}) \to \mathbb{B}$)
Il devient difficile de faire une zoologie détaillée des termes à ce type (ordre 3), mais une pathologie importante apparaît là.

Comment savoir à quelle question répond V dans la trace suivante :

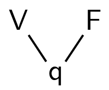
	(($\mathbb{B}$	$\to \mathbb{B}$)	$\to \mathbb{B}$)	$\to \mathbb{B}$
O				q
J			q	
O		q		
J			q	
O		q		
J	q			
O			V	

Ceci correspond à la différence entre les deux λ-termes $\lambda f.(f)\lambda x.(f)\lambda y.x$ et $\lambda f.(f)\lambda x.(f)\lambda y.y$ (termes de *Kierstead*), qui ne sont pas $\beta\eta$-équivalents et que l'on voudrait donc pouvoir différencier. On va voir concrètement qu'il n'est pas possible de les identifier dans le modèle de jeux.

À travers ces différents exemples, on constate que l'on a les contraintes suivantes :
- les traces (parties) commencent toujours par un coup d'Opposant,
- les coups sont alternés (O, J, O, J, ...), globalement mais pas forcément localement,

mais également qu'il faut ajouter d'autres contraintes sur les traces.

Afin de résoudre le problème signalé à la fin de l'exemple 2 et celui de l'exemple 8, on impose que les coups V et F (mais les q à gauche d'une flèche également) soient justifiés par un coup q. Autrement dit, le type $\mathbb{B}$ n'est plus simplement considéré comme un type à trois coups $\{\mathsf{q}, \mathsf{V}, \mathsf{F}\}$ mais avec en plus l'information que q autorise à jouer V et F :

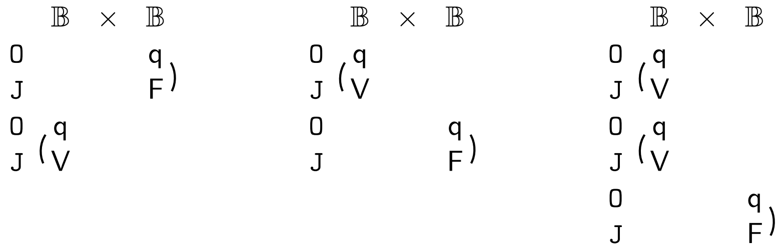


On rajoute également dans les traces un pointeur de chaque coup justifié vers un coup précédent qui le justifie afin de lever l'ambiguïté de l'exemple 8.

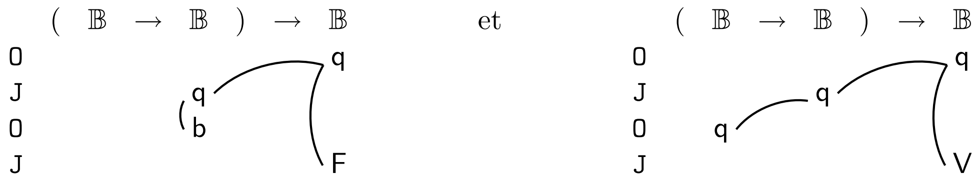
1.1.2 Traces avec pointeurs

On reprend quelques uns des exemples précédents en plaçant explicitement les pointeurs.

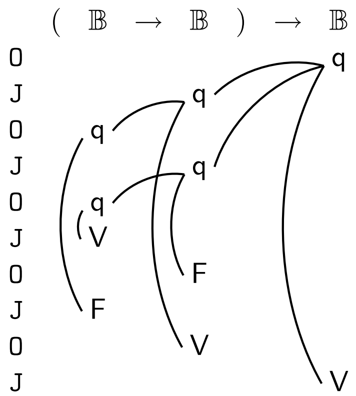
Exemple 9 (Type $\mathbb{B} \times \mathbb{B}$)
Une réponse V ou F pointe sur la question à laquelle elle répond :



Exemple 10 (Type $(\mathbb{B} \to \mathbb{B}) \to \mathbb{B}$)
Une question à gauche d'une flèche pointe sur la question correspondante à droite de la flèche :



Lorsque les appels de fonctions sont emboîtés, les choses se compliquent :



Exemple 11 (Kierstead)

On peut désormais compléter l'exemple 8 avec des pointeurs :

$$
\begin{array}{cccccccccccc}
 & (& (& \mathbb{B} & \to & \mathbb{B} &) & \to & \mathbb{B} &) & \to & \mathbb{B} \\
\mathsf{O} & & & & & & & & & & & \mathsf{q} \\
\mathsf{J} & & & & & & & & \mathsf{q} & & & \\
\mathsf{O} & & & & & \mathsf{q} & & & & & & \\
\mathsf{J} & & & & & & & & \mathsf{q} & & & \\
\mathsf{O} & & & & & \mathsf{q} & & & & & & \\
\mathsf{J} & & & \mathsf{q} & & & & & & & & \\
\mathsf{O} & & & & & & & & \mathsf{V} & & & \\
\mathsf{J} & & & & & \mathsf{V} & & & & & & \mathsf{V}
\end{array}
$$

on commence à voir la différence...

1.1.3 Interactions

Étant donnés un programme de type A et un programme de type $A \to B$, on peut passer le premier comme argument au second pour obtenir un programme de type B. Nous allons décrire comment cette opération est possible au niveau des ensembles de traces.

Exemple 12 (Types $\mathbb{B}$ et $\mathbb{B} \to \mathbb{B}$)
Si on prend le programme de valeur V de type $\mathbb{B}$ et le programme qui représente la négation de type $\mathbb{B} \to \mathbb{B}$, la trace démarre dans le $\mathbb{B}$ de droite de $\mathbb{B} \to \mathbb{B}$ puis on recopie entre le $\mathbb{B}$ de gauche et l'argument les coups successifs, avec un changement du nom du joueur responsable (construction dynamique de l'interaction) :

$$
\begin{array}{ccccccc}
 & \mathbb{B} & & \mathbb{B} & \to & \mathbb{B} & \\
 & & & & & \mathsf{q} & \mathsf{O} \\
\mathsf{O} & \mathsf{q} & \leftarrow & \mathsf{q} & & & \mathsf{J} \\
\mathsf{J} & \mathsf{V} & \rightarrow & \mathsf{V} & & & \mathsf{O} \\
 & & & & & \mathsf{F} & \mathsf{J}
\end{array}
$$

La partie gauche de la trace de $\mathbb{B} \to \mathbb{B}$ est ainsi la même que la trace de l'argument $\mathbb{B}$. Si l'on oublie cette partie commune et que l'on ne garde que la partie droite de la trace de $\mathbb{B} \to \mathbb{B}$, on obtient enfin, comme résultat de la composition, une trace sur $\mathbb{B}$:

$$
\begin{array}{cc}
\mathbb{B} & \\
\mathsf{q} & \mathsf{O} \\
 & \\
 & \\
\mathsf{F} & \mathsf{J}
\end{array}
$$

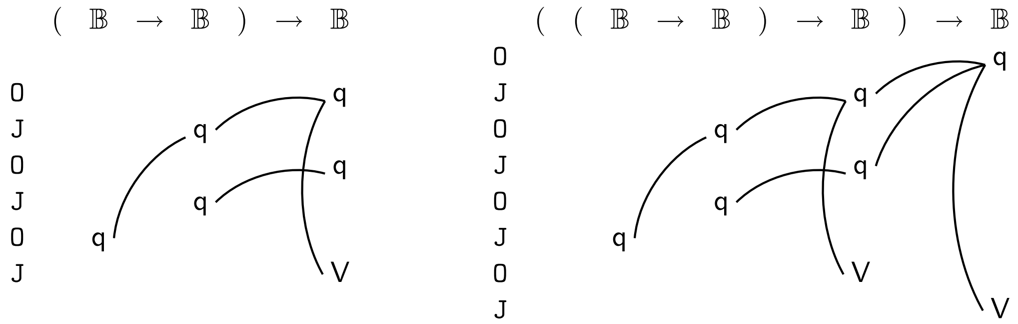
Le résultat de l'application de la négation à l'argument V donne bien la valeur F.

On peut effectuer le même calcul avec une négation qui regarde deux fois son argument :

$$
\begin{array}{ccccccc}
 & \mathbb{B} & & \mathbb{B} & \to & \mathbb{B} & \\
 & & & & & \mathsf{q} & \mathsf{O} \\
\mathsf{O} & \mathsf{q} & \leftarrow & \mathsf{q} & & & \mathsf{J} \\
\mathsf{J} & \mathsf{V} & \rightarrow & \mathsf{V} & & & \mathsf{O} \\
\mathsf{O} & \mathsf{q} & \leftarrow & \mathsf{q} & & & \mathsf{J} \\
\mathsf{J} & \mathsf{V} & \rightarrow & \mathsf{V} & & & \mathsf{O} \\
 & & & & & \mathsf{F} & \mathsf{J}
\end{array}
$$

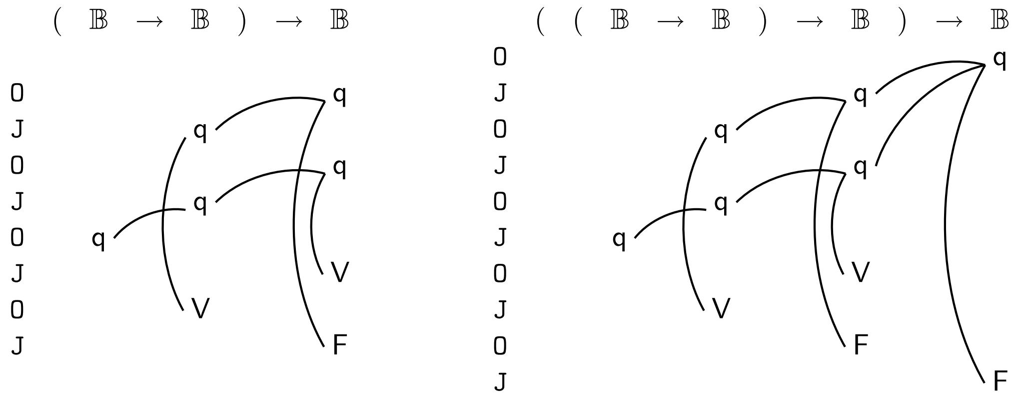
ce qui donne à nouveau la valeur F sur $\mathbb{B}$.

Exemple 13 (Catch et Kierstead)
Les deux traces suivantes sont composables, la première provient de catch et la deuxième du premier terme de *Kierstead* :



et on obtient la valeur booléenne V.

Si, par contre, on compose les traces suivantes, où la première provient toujours de catch mais la deuxième vient de l'autre terme de *Kierstead* :



on obtient la valeur booléenne F. Ce qui montre que l'on a concrètement séparé les deux termes de Kierstead par un contexte applicatif.

1.2 Types et arènes

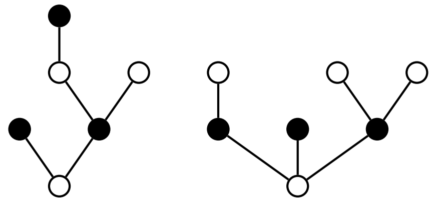
Nous allons désormais formaliser les intuitions que nous avons accumulées à la section précédente.

1.2.1 Définition

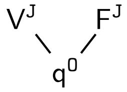
L'interprétation d'un type dans le modèle de jeux est donnée par une arène.

Définition 1 (Arène)
Une *arène* est une forêt finie, dont les nœuds sont appelés *coups*. La *polarité* d'un coup est la parité de la longueur du chemin entre ce coup et la racine de l'arbre auquel il appartient. Les racines sont de polarité O ou $-$, leurs fils sont de polarité J ou $+$, ... Les racines n des arbres sont appelées les *coups initiaux* de l'arène, noté $\vdash n$. Si le coup n est le fils du coup m, on dit que m *valide* n, noté $m \vdash n$.

Exemple 14
Une arène avec deux coups initiaux dont la polarité des coups est écrite explicitement :



Exemple 15 (Arène $\mathbb{B}$)
Nous avons déjà donné informellement l'arène représentant le type $\mathbb{B}$:



1.2.2 Constructions

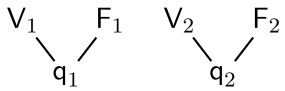
Différentes constructions d'arènes nous seront utiles.

Définition 2 (Arène vide)
La forêt vide est une arène notée $\top$.

Définition 3 (Arène singleton)
La forêt réduite à un seul arbre, lui-même réduit à un seul nœud, est une arène notée $\perp$.

Définition 4 (Arène produit)
Si A et B sont deux arènes, l'arène produit $A \times B$ est l'union disjointe de A et de B.

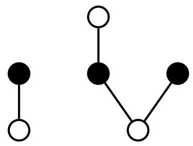
Exemple 16 (Arène $\mathbb{B} \times \mathbb{B}$)
L'arène $\mathbb{B} \times \mathbb{B}$ est obtenue à partir de deux copies disjointes de l'arène $\mathbb{B}$:



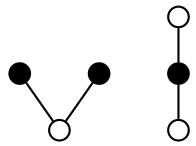
Définition 5 (Arène flèche)
Si A et B sont deux arènes, l'arène flèche $A \to B$ est obtenue en ajoutant chaque racine de A comme fils de chaque racine de B.

Remarque : La polarité des coups provenant de B dans $A \to B$ est la même que celle d'origine, par contre, pour les coups provenant de A, elle est inversée.

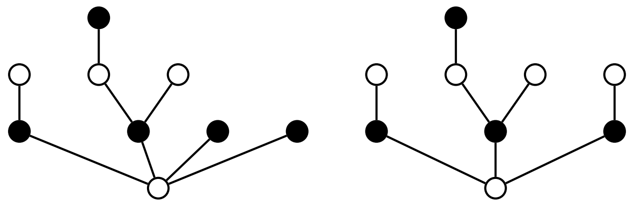
Exemple 17
À partir des arènes A et B suivantes :



et



on obtient l'arène $A \to B$:

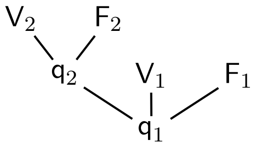


Exercice 2 (Double négation)
Quelle est l'arène $(A \to \bot) \to \bot$? quel effet cette construction a-t-elle sur les coups de A ?

Exercice 3 (Taille de $A \to B$)
Quel est le nombre de coups de $A \to B$, connaissant A et B ? quel est son nombre de coups initiaux ?

Remarque : l'arène $\mathbb{B}$ est la même que celle associée au type $\bot \to \bot \to \bot$.

Exemple 18 (Arène $\mathbb{B} \to \mathbb{B}$)
En "surélevant" une copie de $\mathbb{B}$ au-dessus d'une autre, on obtient l'arène $\mathbb{B} \to \mathbb{B}$:



où les coups q_1, V_2 et F_2 sont de polarité $\mathtt{O}$ et les coups q_2, V_1 et F_1 sont de polarité $\mathtt{J}$.

Remarque : on a noté de la même manière les constructeurs de types et les constructions d'arènes correspondantes (via l'interprétation des types dans les arènes).

Exemple 19
La forêt décrite à l'exemple 14 est l'interprétation du type $(\bot \to ((\bot \to \bot) \to \bot \to \bot) \to \bot) \times ((\bot \to \bot) \to \bot \to (\bot \to \bot \to \bot) \to \bot)$.

Exercice 4 (Quelques égalités)
Si A, B et C sont trois arènes, montrer que :

$$A \times \top = A \tag{1}$$
$$A \to (B \times C) = (A \to B) \times (A \to C) \tag{2}$$
$$(A \times B) \to C = A \to B \to C \tag{3}$$
$$A \to B \to C = B \to A \to C \tag{4}$$

on pourra commencer par le cas où A, B et C sont des arbres.

Exercice 5 (Catégorie d'arènes)
Montrer qu'en prenant comme objets les arènes et comme morphismes les morphismes de graphes, on peut construire une catégorie. Quelles sont les propriétés de $\top$, $\bot$, $\times$ et $\to$ dans cette catégorie ?

Lemme 1 (Décomposition des arbres)
*Toute arène avec un seul coup initial (*i.e. *tout arbre) est l'interprétation d'un type formé avec $\bot$ et $\to$.*

Démonstration : Soit A une arène à un seul coup initial, on raisonne par récurrence sur la profondeur de A. Si A est de profondeur 1, alors A est réduite à un nœud et $A = \bot$. Si A est de profondeur $n+1$, soient a_1, ..., a_n les fils de la racine de A. La sous-arène de A au-dessus de a_i est un arbre, donc par hypothèse de récurrence, c'est l'interprétation d'un type A_i construit avec $\bot$ et $\to$. On voit alors que A est l'interprétation de $A_1 \to \cdots \to A_n \to \bot$. □

Proposition 1 (Décomposition des arènes)
Toute arène non vide est l'interprétation d'un type formé avec $\bot$, $\to$ et $\times$.

Démonstration : Soient A_1, ..., A_n les arbres de l'arène non vide A, par le lemme 1, les A_i sont l'interprétation de types formés avec $\bot$ et $\to$. Or A s'écrit $A = A_1 \times \cdots \times A_n$ ce qui permet de conclure. □

1.3 Stratégies

L'interprétation d'un terme dans le modèle de jeux est donnée par une stratégie sur l'arène correspondante.

1.3.1 Définition

Définition 6 (Suite pointée)
Une *suite pointée* sur l'arène A est un couple (s, f) où s est une suite finie de coups de A et, si k est la longueur $|s|$ de s ($s = n_1 n_2 ... n_k$), f est une fonction de $\{1, ..., k\}$ dans $\{0, ..., k-1\}$ telle que :
- $f(i) < i$
- si $f(i) = 0$ alors n_i est un coup initial de A
- si $f(i) = j \neq 0$ alors n_j valide n_i dans A (on dit que n_j *justifie* n_i dans (s, f) ou que n_i *pointe* sur n_j).

S'il existe $p > 0$ tel que $f^p(i) = j \neq 0$, on dit que n_j justifie *héréditairement* n_i dans (s, f).

Remarque : Soit (s, f) une suite pointée non vide sur l'arène A, le premier coup de s est un coup d'Opposant. En effet on doit avoir $f(1) < 1$ donc $f(1) = 0$ ce qui entraîne que le premier coup de s est un coup initial donc un coup d'Opposant.

Définition 7 (Partie)
Une *partie* sur l'arène A est une suite pointée sur A dont la polarité des coups est alternée.

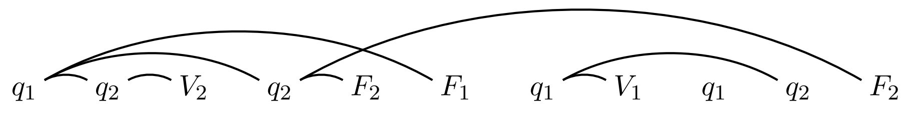
L'ensemble des parties sur l'arène A est noté $\mathcal{P}_A$.

Remarque : Dans une partie (s, f), si i est pair (resp. impair), $f(i)$ est impair (resp. pair).

Exemple 20
Sur l'arène $\mathbb{B} \to \mathbb{B}$, si $s = q_1\, q_2\, V_2\, q_2\, F_2\, F_1\, q_1\, V_1\, q_1\, q_2\, F_2$ (avec les notations de l'exemple 18) et

$$f : \begin{array}{rcl} 1 & \mapsto & 0 \\ 2 & \mapsto & 1 \\ 3 & \mapsto & 2 \\ 4 & \mapsto & 1 \\ 5 & \mapsto & 4 \\ 6 & \mapsto & 1 \\ 7 & \mapsto & 0 \\ 8 & \mapsto & 7 \\ 9 & \mapsto & 0 \\ 10 & \mapsto & 7 \\ 11 & \mapsto & 4 \end{array}$$

alors (s, f) est une partie que l'on représentera par :



Remarque : Toute arène possède au moins une partie : la partie vide notée ε.

Exercice 6
Montrer que dans une partie (s, f) qui ne contient qu'un seul coup initial, les valeurs de $f(1)$, $f(2)$ et $f(3)$ sont déterminées de manière unique. Que peut-on dire s'il y a plusieurs coups initiaux ?

Maintenant que l'on a donné une définition rigoureuse, on utilisera toujours la représentation avec pointeurs et on identifiera souvent un coup et une occurrence de coup dans une partie (resp. suite pointée), ou un coup et un coup avec son pointeur. On notera désormais s les parties (resp. suites pointées) et non plus (s, f).

Définition 8 (Projection)
Soit s une suite pointée de $A \times B$ (ou $A \to B$), la *projection* de s sur A (resp. B), notée $s \upharpoonright_A$ (resp. $s \upharpoonright_B$), est la sous-suite pointée de s obtenue en ne gardant que les coups de A (resp. B).

Remarque : Même si s est une partie, $s \upharpoonright_A$ n'est pas en général une partie de A :

$$\begin{array}{cccc} & \mathbb{B} & \to & \mathbb{B} \\ \mathtt{O} & & & \mathtt{q} \\ \mathtt{J} & \mathtt{q} & & \\ \mathtt{O} & & & \mathtt{q} \\ \mathtt{J} & \mathtt{q} & & \end{array}$$

Définition 9 (Stratégie)
Une *stratégie* σ sur l'arène A, noté $\sigma : A$, est un ensemble non vide de parties sur l'arène A de longueur paire (appelées J-parties) clos par préfixes (l'ordre préfixe est noté $\leq$) de longueur paire (ou J-préfixes) tel que :
si $sm \in \sigma$ et $sn \in \sigma$ alors $m = n$ et ils pointent sur le même coup (déterminisme).

Remarque : Sur toute arène A, il existe une stratégie particulière appelée *stratégie vide* : $\{\varepsilon\}$.

Définition 10 (Stratégie – représentation fonctionnelle)
Une *stratégie* σ sur l'arène A est une fonction partielle des parties sur A de longueur impaire (ou O-parties) dans les coups de Joueur de A telle que si $f(n_1 ... n_{2k+1}) = m$ alors pour tout $i < k$, $f(n_1 ... n_{2i+1}) = n_{2i+2}$.

Exercice 7
Montrer qu'il y a une correspondance bijective entre les deux définitions de stratégie.

Exercice 8
Formaliser avec des arènes et des stratégies les exemples donnés à la section 1.1.

1.3.2 Catégorie

On va construire une catégorie dont les objets sont les arènes et dont les morphismes de A dans B sont les stratégies sur $A \to B$.

Définition 11 (Stratégie identité)
Soit A une arène, la stratégie id_A sur l'arène $A \to A$ est définie par :

$$id_A = \{s \in \mathcal{P}_{A_1 \to A_2} \mid |s| \text{ paire} \wedge \forall t \leq s, |t| \text{ paire}, t \upharpoonright_{A_1} = t \upharpoonright_{A_2}\}$$

où les indices ne servent qu'à distinguer les deux occurrences de A.

Lemme 2
id_A est une stratégie sur $A \to A$.

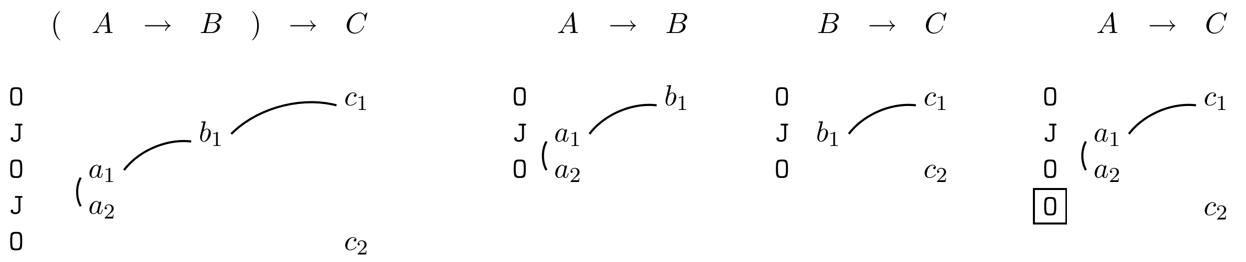
Démonstration : On a quatre propriétés à vérifier :

- id_A est non vide puisque $\varepsilon \in id_A$;
- toute $s \in id_A$ est de longueur paire par définition ;
- si $s \in id_A$ et $t \leq s$ de longueur paire alors $t \in id_A$ par définition ;
- si $sm \in id_A$ et $sn \in id_A$ alors $sm \upharpoonright_{A_1} = sm \upharpoonright_{A_2}$ et $sn \upharpoonright_{A_1} = sn \upharpoonright_{A_2}$ mais m ne peut pas être à la fois dans A_1 et dans A_2, s'il est dans A_1 on a $sm \upharpoonright_{A_2} = s \upharpoonright_{A_2}$. Par un argument de longueur, si n était dans A_2, on aurait $|s \upharpoonright_{A_1}| = |sn \upharpoonright_{A_1}| = |sn \upharpoonright_{A_2}| = |s \upharpoonright_{A_2}| + 1$ et de même avec m, $|s \upharpoonright_{A_2}| = |s \upharpoonright_{A_1}| + 1$. On en déduit que n est nécessairement dans la même composante que m et on a finalement : $sm \upharpoonright_{A_1} = sm \upharpoonright_{A_2} = s \upharpoonright_{A_2} = sn \upharpoonright_{A_2} = sn \upharpoonright_{A_1}$ d'où $m = n$. □

Définition 12 (Interaction)
Soient A, B et C trois arènes, une *interaction* u sur A, B, C, noté $u \in \mathrm{int}(A, B, C)$, est une suite pointée de $(A \to B) \to C$ telle que $u \upharpoonright_{A,B}$, $u \upharpoonright_{B,C}$ et $u \upharpoonright_{A,C}$ sont des parties sur $A \to B$, $B \to C$ et $A \to C$. Concernant les notations :

- $u \upharpoonright_{A,B} = u \upharpoonright_{A \to B}$,
- $u \upharpoonright_{B,C}$ est la sous-suite pointée de u obtenue en ne gardant que les coups de B et C,
- $u \upharpoonright_{A,C}$ est la sous-suite pointée de u obtenue en ne gardant que les coups de A et C, et pour tout coup a dans A qui pointe sur b dans B (ce qui implique que b est initial dans B par définition de $A \to B$) et qui pointe lui-même sur c dans C (initial dans C par définition de $(A \to B) \to C$) on fait pointer a sur c dans $u \upharpoonright_{A,C}$ (ce qui est correct puisque c valide a dans $A \to C$).

Remarque : Il est nécessaire d'imposer les contraintes sur les projections, car on peut avoir une suite pointée s de $(A \to B) \to C$ telle que $u \upharpoonright_{A,B}$ et $u \upharpoonright_{B,C}$ sont des parties mais $u \upharpoonright_{A,C}$ n'est pas alternée :



Une information utile à extraire de la définition d'interaction est l'ensemble des contraintes que cela impose sur l'ordre des coups dans A, B et C. Soit $u \in \mathrm{int}(A, B, C)$, on note tout d'abord qu'on a la correspondance suivante entre la polarité p d'un coup dans u et sa polarité dans les trois projections de u selon la composante dans laquelle il se trouve :

	$u \upharpoonright_{A,B}$	$u \upharpoonright_{B,C}$	$u \upharpoonright_{A,C}$
A	$\neg p$		$\neg p$
B	$\neg p$	p	
C		p	p

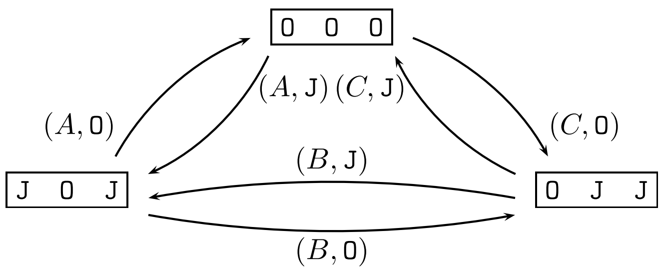
On appelle état de u le triplet (p_1, p_2, p_3) des polarités du coup à venir dans chacune des projections, qui doivent être alternées par définition d'une interaction. Ainsi, l'état de l'interaction vide $u = \varepsilon$

est $(\mathtt{O},\mathtt{O},\mathtt{O})$. On va construire l'*automate des états* avec comme transitions les paires (X, J) où X est l'une des trois composantes A, B ou C, et J est le joueur qui joue : une arête étiquetée (X, J) entre l'état e_1 et l'état e_2 signifie que si le joueur J joue un coup dans X alors que l'interaction est dans l'état e_1, on obtient une interaction dans l'état e_2.

On traite, en détails, le cas des transitions partant de l'état $(\mathtt{O},\mathtt{O},\mathtt{O})$:

- une transition $(A, \mathtt{O})$ correspond à un coup J dans $u \upharpoonright_{A,B}$, impossible,
- une transition $(B, \mathtt{O})$ correspond à un coup J dans $u \upharpoonright_{A,B}$, impossible,
- une transition $(C, \mathtt{O})$ correspond à un coup O dans $u \upharpoonright_{B,C}$ et $u \upharpoonright_{A,C}$, elle est donc valide et mène à l'état $(\mathtt{O},\mathtt{J},\mathtt{J})$,
- une transition $(A, \mathtt{J})$ correspond à un coup O dans $u \upharpoonright_{A,B}$ et $u \upharpoonright_{A,C}$, elle est donc valide et mène à l'état $(\mathtt{J},\mathtt{O},\mathtt{J})$,
- une transition $(B, \mathtt{J})$ correspond à un coup J dans $u \upharpoonright_{B,C}$, impossible,
- une transition $(C, \mathtt{J})$ correspond à un coup J dans $u \upharpoonright_{B,C}$, impossible.

Ce qui donne (en complétant le calcul) l'automate suivant :



On peut lire différentes informations sur cet automate :

- un coup J dans A (resp. O dans C) ne peut pas être suivi d'un coup dans C (resp. A) ;
- si $u \upharpoonright_{A,B}$ et $u \upharpoonright_{B,C}$ sont de longueur paire (état de la forme $(\mathtt{O},\mathtt{O},p)$) alors $u \upharpoonright_{A,C}$ est de longueur paire (le seul tel état est $(\mathtt{O},\mathtt{O},\mathtt{O})$).

Définition 13 (Composition)
Si $\sigma : A \to B$ et $\tau : B \to C$ sont deux stratégies, la composée $\sigma;\tau$ est obtenue par :

$$\sigma;\tau = \{u \upharpoonright_{A,C} \mid u \in \mathrm{int}(A, B, C), u \upharpoonright_{A,B} \in \sigma, u \upharpoonright_{B,C} \in \tau\}$$

Remarque : On note la composition dans le sens inverse de la composition usuelle des fonctions pour faciliter les écritures : $\sigma;\tau = \tau \circ \sigma$.

Lemme 3 (Zipping)
Soient s et t deux parties de $\sigma;\tau$ ayant les mêmes coups d'Opposant (et mêmes pointeurs pour ces coups), alors $s = t$.

Démonstration : Soit u une interaction telle que $u \upharpoonright_{A,C} = s$ et v une interaction telle que $v \upharpoonright_{A,C} = t$. On va montrer que si u et v ont les mêmes coups J dans A et les mêmes coups O dans C alors $u \leq v$ ou $v \leq u$, avec uniquement des coups dans B dans la partie restante et éventuellement un coup O dans A ou J dans C (ce qui implique que $s = t$, car elles ne peuvent pas différer d'un seul coup puisqu'elles sont de longueur paire).

Soit w le préfixe de longueur maximale commun à u et v, si $|w| < \inf(|u|, |v|)$, on a $u = wa...$ et $v = wb....$ On considère les différents états possibles de w, et on applique l'automate des états :

- Si w est dans l'état $(\mathtt{O},\mathtt{O},\mathtt{O})$, a et b sont des coups $(A, \mathtt{J})$ ou $(C, \mathtt{O})$, mais par hypothèse, ces coups sont les mêmes dans u et dans v donc $a = b$ et w n'est pas un préfixe commun de longueur maximale.

- Si w est dans l'état $(\mathtt{J}, \mathtt{O}, \mathtt{J})$, a et b sont des coups $(A, \mathtt{O})$ ou $(B, \mathtt{O})$, donc ce sont des coups $(A, \mathtt{J})$ ou $(B, \mathtt{J})$ dans $u \upharpoonright_{A,B} \in \sigma$ et $v \upharpoonright_{A,B} \in \sigma$ ce qui entraîne que $a = b$ par déterminisme de σ, et donc w n'est pas un préfixe commun de longueur maximale.
- Si w est dans l'état $(\mathtt{O}, \mathtt{J}, \mathtt{J})$, par déterminisme de τ, w ne peut pas être un préfixe commun de longueur maximale.

On en déduit que $w = u$ ou $w = v$ or u et v ont les mêmes coups J dans A et O dans C, puisque l'un est préfixe de l'autre, la partie restante de la plus longue ne peut contenir que des coups dans B, des coups O dans A et des coups J dans C, mais il y a au plus un tel coup dans A ou C car d'après l'automate des états il doit être suivi par un coup J dans A ou un coup O dans C. □

Lemme 4
Si $\sigma : A \to B$ et $\tau : B \to C$ alors $\sigma; \tau : A \to C$.

Démonstration : On doit montrer que $\sigma; \tau$ est bien une stratégie :
- $\sigma; \tau$ est non vide puisque $\varepsilon \in \mathrm{int}(A, B, C)$, $\varepsilon \upharpoonright_{A,B} = \varepsilon \in \sigma$, $\varepsilon \upharpoonright_{B,C} = \varepsilon \in \tau$, et $\varepsilon \upharpoonright_{A,C} = \varepsilon$ donc $\varepsilon \in \sigma; \tau$.
- $\sigma; \tau$ est un ensemble de parties par définition de $\mathrm{int}(A, B, C)$.
- Si $u \in \mathrm{int}(A, B, C)$ et $u \upharpoonright_{A,B} \in \sigma$, $u \upharpoonright_{B,C} \in \tau$ alors par définition d'une stratégie $u \upharpoonright_{A,B}$ et $u \upharpoonright_{B,C}$ sont de longueur paire. D'après la remarque ci-dessus qui suit l'automate des états, $u \upharpoonright_{A,C}$ est également de longueur paire.
- Par définition, toute partie de $\sigma; \tau$ est de la forme $u \upharpoonright_{A,C}$ avec $u \in \mathrm{int}(A, B, C)$. Soit v le préfixe maximal de u tel qu'il existe exactement deux coups de A ou C qui soient dans u et pas dans v. On a $v \in \mathrm{int}(A, B, C)$, $v \upharpoonright_{A,B} \in \sigma$ car $v \upharpoonright_{A,B} \leq u \upharpoonright_{A,B}$ et $|v \upharpoonright_{A,B}|$ paire (par l'automate des états), $v \upharpoonright_{B,C} \in \tau$ car $v \upharpoonright_{B,C} \leq u \upharpoonright_{B,C}$ et $|v \upharpoonright_{B,C}|$ paire donc $v \upharpoonright_{A,C} \in \sigma; \tau$. On vérifie alors que $v \upharpoonright_{A,C}$ est le préfixe de $u \upharpoonright_{A,C}$ de longueur diminuée de 2. On continue de même par récurrence pour les autres préfixes de longueur paire.
- Si $sm^{\mathtt{J}} \in \sigma; \tau$ et $sn^{\mathtt{J}} \in \sigma; \tau$ alors sm et sn ont les mêmes coups O donc, par le lemme 3, $sm = sn$. □

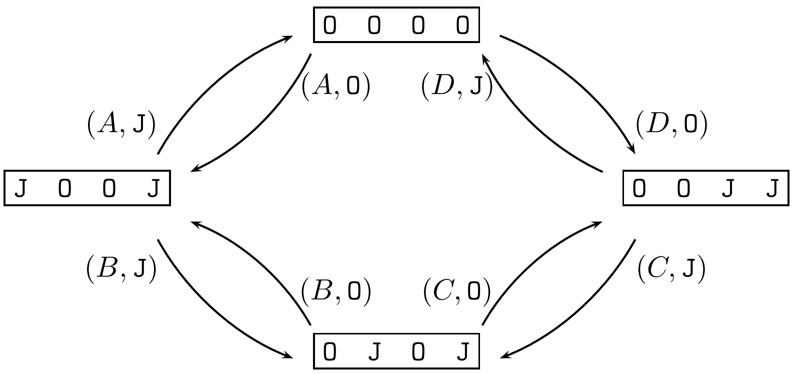
Définition 14 (Interaction double)
Soient A, B, C et D quatre arènes, une *interaction double* w sur A, B, C, D, noté $w \in \mathrm{int}(A, B, C, D)$, est une suite pointée de $((A \to B) \to C) \to D$ telle que $w \upharpoonright_{A,B}$, $w \upharpoonright_{B,C}$, $w \upharpoonright_{C,D}$ et $w \upharpoonright_{A,D}$ sont des parties sur $A \to B$, $B \to C$, $C \to D$ et $A \to D$.

Les notations pour les projections sont la généralisation immédiate de celles pour les interactions simples (définition 12).

On construit un automate des états pour les interactions doubles. Cette fois un état est le quadruplet des polarités dans $A \to B$, $B \to C$, $C \to D$ et $A \to D$. Les polarités relatives dans les projections sont obtenues par :

	$w \upharpoonright_{A,B}$	$w \upharpoonright_{B,C}$	$w \upharpoonright_{C,D}$	$w \upharpoonright_{A,D}$
A	p			p
B	p	$\neg p$		
C		$\neg p$	p	
D			p	p

L'automate des états se calcule comme dans le cas simple :



Lemme 5
Si $w \in \mathrm{int}(A, B, C, D)$ *alors* $w \restriction_{A,C} \in \mathcal{P}_{A \to C}$ *et* $w \restriction_{B,D} \in \mathcal{P}_{B \to D}$.

Démonstration : On traite le cas de B et D (celui de A et C étant parfaitement similaire). Les polarités dans $B \to D$ sont otenues en niant celle de B et conservant celle de D, on doit donc montrer que si on a coup $(B, \mathtt{O})$ (resp. $(B, \mathtt{J})$) alors le coup suivant dans B ou D est un coup $(B, \mathtt{J})$ ou un coup $(D, \mathtt{O})$ (resp. $(B, \mathtt{O})$ ou $(D, \mathtt{J})$). En regardant sur l'automate des états, un coup $(B, \mathtt{O})$ mène à l'état $(\mathtt{J}, \mathtt{O}, \mathtt{O}, \mathtt{J})$ et tant que l'on ne joue que dans A ou dans C, on reste dans les états $(\mathtt{J}, \mathtt{O}, \mathtt{O}, \mathtt{J})$ et $(\mathtt{O}, \mathtt{O}, \mathtt{O}, \mathtt{O})$, ne peut en sortir que par un coup $(B, \mathtt{J})$ ou un coup $(D, \mathtt{O})$. Le cas de D se traite de la même manière.

Remarque : Les contraintes sur les projections sont ainsi équivalentes à demander que $w \restriction_{A,B,C} \in \mathrm{int}(A, B, C)$ et $w \restriction_{A,C,D} \in \mathrm{int}(A, C, D)$ ou que $w \restriction_{A,B,D} \in \mathrm{int}(A, B, D)$ et $w \restriction_{B,C,D} \in \mathrm{int}(B, C, D)$.

Lemme 6 (Zipping double)
Soient $u \in \mathrm{int}(A, B, C)$ *et* $v \in \mathrm{int}(A, C, D)$ *telles que* $u \restriction_{A,C} = v \restriction_{A,C}$*, il existe* $w \in \mathrm{int}(A, B, C, D)$ *telle que* $w \restriction_{A,B,C} = u$ *et* $w \restriction_{A,C,D} = v$.

Démonstration : Par récurrence sur $|u \restriction_{A,C}|$:

- Si $|u \restriction_{A,C}| = 0$ alors $u = \varepsilon$ et v ne contient que des coups dans C et D, on peut prendre $w = v$.
- Si $|u \restriction_{A,C}| = u'n$ on va montrer que n est le dernier coup de u ou de v. Si ce n'est pas le cas, il est suivi par un coup de B dans u et par un coup de D dans v. Supposons que n est dans C, par les automates des états pour u et v, n doit être un coup $\mathtt{O}$ dans u (donc $\mathtt{O}$ dans $A \to C$) et un coup $\mathtt{O}$ dans v (donc $\mathtt{J}$ dans $A \to C$) ce qui est impossible, et de même si n est dans A.
  Supposons que n est le dernier coup de u. On peut écrire $u = u'n$ et $v = v'nv''$ avec v'' formé uniquement de coups de D. Par hypothèse de récurrence, il existe $w' \in \mathrm{int}(A, B, C, D)$ telle que $w' \restriction_{A,B,C} = u'$ et $w' \restriction_{A,C,D} = v'$. On pose $w = w'nv''$, il faut montrer que $w \in \mathrm{int}(A, B, C, D)$. On a $w \restriction_{A,B,C} = w'nv'' \restriction_{A,B,C} = w'n \restriction_{A,B,C} = u$ est une interaction sur A, B, C et $w \restriction_{A,C,D} = w'nv'' \restriction_{A,C,D} = v'nv'' = v$ est aussi une interaction sur A, C, D, ce qui permet de conclure. □

Lemme 7 (Associativité)
Si $\sigma : A \to B$*,* $\tau : B \to C$ *et* $\rho : C \to D$ *alors* $(\sigma; \tau); \rho = \sigma; (\tau; \rho)$.

Démonstration : On va se contenter de montrer la première inclusion, la seconde s'obtenant de manière très similaire.

Si $s \in (\sigma;\tau);\rho$, il existe une interaction $v \in \mathrm{int}(A, C, D)$ telle que $v \restriction_{A,C} \in \sigma;\tau$, $v \restriction_{C,D} \in \rho$ et $v \restriction_{A,D} = s$. De même il existe une interaction $u \in \mathrm{int}(A, B, C)$ telle que $u \restriction_{A,B} \in \sigma$, $u \restriction_{B,C} \in \tau$ et $u \restriction_{A,C} = v \restriction_{A,C}$. Par le lemme 6, il existe $w \in \mathrm{int}(A, B, C, D)$ telle que $w \restriction_{A,B,C} = u$ et $w \restriction_{A,C,D} = v$. Pour montrer que $s \in \sigma;(\tau;\rho)$ il suffit de montrer que $w \restriction_{A,B,D} \in \mathrm{int}(A, B, D)$ avec $w \restriction_{A,B} \in \sigma$ et $w \restriction_{B,D} \in \tau;\rho$ puisque $w \restriction_{A,D} = v \restriction_{A,D} = s$. On sait déjà que $w \restriction_{A,B} = u \restriction_{A,B} \in \sigma$, il reste donc à prouver que $w \restriction_{B,D} \in \tau;\rho$. Pour cela, on a $w \restriction_{B,C} = u \restriction_{B,C} \in \tau$, $w \restriction_{C,D} = v \restriction_{C,D} \in \rho$, et $w \restriction_{B,D} \in \mathcal{P}_{B\to D}$ par le lemme 5. □

Lemme 8 (Neutralité)
Si $\sigma : A \to B$ alors $id_A;\sigma = \sigma$ et $\sigma; id_B = \sigma$.

Démonstration : On va se contenter de prouver la première inclusion de la première égalité.

Si $s \in id_A;\sigma$, il existe une interaction u telle que $u \restriction_{A_1,B} = s$, $u \restriction_{A_1} = u \restriction_{A_2}$ et $u \restriction_{A_2,C} \in \sigma$. On a donc $u \restriction_{A_2,C} = s$ et $u \restriction_{A_2,C} \in \sigma$ d'où $s \in \sigma$. □

Proposition 2 (Catégorie de jeux)
En prenant comme objets les arènes, comme morphismes de A dans B les stratégies sur $A \to B$, comme identités les stratégies id et comme loi de composition ";", on obtient une catégorie notée $\mathcal{G}$.

Remarque : Toute stratégie $\sigma : A$ peut être vue comme une stratégie $\sigma : \top \to A$ (et réciproquement). Si $\tau : A \to B$, on notera $\sigma;\tau : B$ la stratégie obtenue en composant $\top \xrightarrow{\sigma} A$ avec $A \xrightarrow{\tau} B$ ce qui donne une stratégie sur $\top \to B$ qui peut à son tour être vue comme une stratégie sur B.

1.3.3 Constructions

On cherche à étudier la structure de la catégorie $\mathcal{G}$ afin de voir de quel langage elle fournit un modèle. On définit tout d'abord quelques constructions dans cette catégorie.

Objet terminal. L'arène $\top$ est un objet terminal. En effet si A est une arène quelconque et $\sigma : A \to \top$ une stratégie, $A \to \top$ étant l'arène vide, on a $\sigma = \{\varepsilon\}$. On notera $\varepsilon_A : A \to \top$ cette stratégie.

Produit. Soient A et B deux arènes, les projections π_A et π_B de $A \times B$ dans A et B sont définies par :

$$\pi_A = \{s \in \mathcal{P}_{A\times B\to A} \mid s \in id_A\}$$

où $s \in id_A$ signifie que s ne comporte pas de coup dans B et donc qu'elle peut être vue comme une partie sur $A \to A$ à laquelle on impose d'être dans id_A. On définit π_B de la même manière.

Lemme 9 (Stratégies de projection)
π_A et π_B sont des stratégies.

Si $\sigma : A \to C$ et $\tau : B \to D$ sont deux stratégies, on définit la stratégie $\sigma \times \tau : A \times B \to C \times D$ par :

$$\sigma \times \tau = \{s \in \mathcal{P}_{A\times B\to C\times D} \mid s \restriction_{A\to C} \in \sigma \wedge s \restriction_{B\to D} \in \tau\}$$

où les projections $s \restriction_{A\to C}$ et $s \restriction_{B\to D}$ sont obtenues de la manière naturelle en ne gardant dans s que les coups provenant des bonnes composantes.

Lemme 10 (Stratégie produit)
Si $\sigma : A \to C$ et $\tau : B \to D$ sont des stratégies, $\sigma \times \tau : A \times B \to C \times D$ est une stratégie.

Démonstration : On vérifie les quatre propriétés d'une stratégie :
- $\sigma \times \tau$ n'est pas vide puisque $\varepsilon \in \sigma \times \tau$;
- les éléments de $\sigma \times \tau$ sont de longueur paire puisque les deux projections (disjointes) le sont ;
- $\sigma \times \tau$ est clos pas préfixe pair par définition ;
- si $sm \in \sigma \times \tau$ et $sn \in \sigma \times \tau$, on montre par un automate des états qu'un coup `O` suivi d'un coup `J` dans une partie de $\sigma \times \tau$ ne peuvent pas être l'un dans A ou C et l'autre dans B ou D. En regardant dans quelle composante se trouve le dernier coup de s, on peut donc en déduire que m et n sont tous les deux dans A et C ou dans B et D, d'où on conclut par déterminisme de σ et τ. $\square$

Exercice 9
Montrer que $\pi_A = id_A \times \varepsilon_B$.

Proposition 3 (Bifoncteur $\times$)
La construction $\times$ est un bifoncteur de $\mathcal{G}$ et $\mathcal{G}$ vers $\mathcal{G}$.

Exercice 10
Prouver la proposition 3.

Définition 15 (Diagonale)
Soit A une arène, la stratégie *diagonale* sur $A \to A \times A$, notée Δ_A, est définie par :

$$\Delta_A = \{s \in \mathcal{P}_{A_0 \to A_1 \times A_2} \mid s \upharpoonright_{A_0^1 \to A_1} \in id_A \wedge s \upharpoonright_{A_0^2 \to A_2} \in id_A\}$$

où les indices ne servent qu'à distinguer des occurrences de A et où $s \upharpoonright_{A_0^i \to A_i}$ est la sous-suite pointée de s obtenue en gardant les coups de A_i et les coups de A_0 héréditairement justifiés par un coup initial dans A_i.

Exercice 11 (Diagonale et pointeurs)
Montrer que Δ est une stratégie sur $A \to A \times A$. Montrer pourquoi la définition de Δ poserait un problème en l'absence de pointeurs.

Le produit n'est pas cartésien. Malheureusement on ne va pas pouvoir montrer que la catégorie $\mathcal{G}$ est cartésienne fermée...

Lemme 11 (Produit faible)
Soient $\sigma : C \to A$ et $\tau : C \to B$ deux stratégies, il existe une stratégie $\rho : C \to A \times B$ qui fait commuter le diagramme :

$$\begin{tikzcd}
 & A \\
C \arrow[ur, "\sigma"] \arrow[r, dashed, "\rho"] \arrow[dr, "\tau"'] & A \times B \arrow[u, "\pi_A"'] \arrow[d, "\pi_B"] \\
 & B
\end{tikzcd}$$

Démonstration : On pose $\rho = \Delta_C; (\sigma \times \tau)$ et on montre que le diagramme commute. On se contente du triangle supérieur, l'autre se prouvant exactement de la même manière. À l'aide de l'exercice 9, de la proposition 3 et du fait que $\top$ est un objet terminal, on montre :

$$\begin{aligned}\Delta_C; (\sigma \times \tau); \pi_A &= \Delta_C; (\sigma \times \tau); (id_A \times \varepsilon_B) \\ &= \Delta_C; ((\sigma; id_A) \times (\tau; \varepsilon_B)) \\ &= \Delta_C; (\sigma \times (\tau; \varepsilon_B)) \\ &= \Delta_C; (\sigma \times \varepsilon_C) \\ &= \Delta_C; (id_C \times \varepsilon_C); \sigma\end{aligned}$$

On montre alors que $\Delta_C; (id_C \times \varepsilon_C) = id_C$ car tout coup joué dans $C \times C$ par $(id_C \times \varepsilon_C)$ est joué dans la composante gauche. $\square$

Remarque : On note $\langle \sigma, \tau \rangle : C \to A \times B$ la stratégie $\Delta_C; (\sigma \times \tau)$ utilisée ci-dessus.

Exercice 12
Montrer que l'on peut définir $\langle \sigma, \tau \rangle$ par :

$$\langle \sigma, \tau \rangle = \{ s \in \mathcal{P}_{C \to A \times B} \mid s \upharpoonright_{C^A \to A} \in \sigma \wedge s \upharpoonright_{C^B \to B} \in \tau \}$$

avec le même genre de notations que dans la définition 15.

L'unicité de ρ dans le lemme 11, qui ferait de $(A \times B, \pi_A, \pi_B)$ un produit cartésien sur $\mathcal{G}$, n'est pas vérifiée. On considère les stratégies identité $id_{\mathbb{B}} : \mathbb{B} \to \mathbb{B}$ et négation $\neg : \mathbb{B} \to \mathbb{B}$. La stratégie $\langle id_{\mathbb{B}}, \neg \rangle : \mathbb{B} \to \mathbb{B} \times \mathbb{B}$ fait commuter le diagramme de produit comme on l'a vu. Mais la stratégie obtenue en modifiant $\langle id_{\mathbb{B}}, \neg \rangle$ pour qu'elle contienne la partie :

$$\begin{array}{cccccc} & \mathbb{B} & \to & \mathbb{B} & \times & \mathbb{B} \\ \mathsf{O} & & & \mathsf{q} & & \\ \mathsf{J} & \mathsf{q} & & & & \\ \mathsf{O} & \mathsf{V} & & & & \\ \mathsf{J} & & & \mathsf{V} & & \\ \mathsf{O} & & & & & \mathsf{q} \\ \mathsf{J} & & & & & \mathsf{V} \end{array}$$

fait également commuter ce diagramme !

Flèche. La catégorie $\mathcal{G}$ possède également des exponentiations.

Définition 16 (Foncteur $A \to \_$)
Soit A une arène, le foncteur $A \to \_$ est défini par :
- si B est une arène $A \to \_(B)$ est l'arène $A \to B$;
- si σ est un morphisme de B dans C (donc une stratégie sur $B \to C$), $A \to \_(\sigma)$ est la stratégie :

$$A \to \sigma = \{ s \in \mathcal{P}_{(A \to B) \to (A \to C)} \mid s \upharpoonright_{A \to A} \in id_A \wedge s \upharpoonright_{B \to C} \in \sigma \}$$

Remarque : Dans la catégorie $\mathcal{G}$, les arènes $A \times B \to C$ et $A \to B \to C$ sont les mêmes donc $\mathcal{G}(A \times B, C) = \mathcal{G}(A, B \to C)$.

Exercice 13
Montrer que $A \to \_$ est un foncteur de $\mathcal{G}$ dans $\mathcal{G}$. Montrer que l'on peut étendre la définition de $\to$ pour obtenir un bifoncteur de $\mathcal{G}^{\mathrm{op}}$ et $\mathcal{G}$ dans $\mathcal{G}$.

Exercice 14 (SMCC)
Montrer que $(\mathcal{G}, \times, \top, \to)$ donne une catégorie symétrique monoïdale fermée.

Remarque : Puisque $\mathcal{G}$ n'est pas une CCC, on ne peut pas appliquer le résultat général concernant les modèles catégoriques du λ-calcul. Cependant on a obtenu presque toute la structure nécessaire pour une CCC et on peut montrer que c'est suffisant pour donner une interprétation du λ-calcul dans $\mathcal{G}$ qui en fasse un modèle dénotationnel.

1.3.4 Stratégies filaires

Une manière d'énoncer le problème que l'on a rencontré avec le produit dans $\mathcal{G}$ est de dire qu'il existe des pseudo-paires qui font commuter le diagramme de produit. On élimine ces pseudo-paires en ajoutant de nouvelles contraintes sur les stratégies de manière à ne garder *qu'une* vraie stratégie paire.

Définition 17 (Fil)
Soit s une partie de l'arène A, le *fil* $\lceil s \rceil$ de s est la sous-suite pointée de s contenant tous les coups héréditairement justifiés par le coup initial de s qui justifie héréditairement le dernier coup de s.

Remarque : Le fil d'une partie s contient exactement un coup initial (si s est non vide). On appellera *fil* toute suite pointée qui est vide ou qui possède un unique coup initial. Un partie s est un fil *ssi* $s = \lceil s \rceil$.

Définition 18 (Stratégie filaire)
Une stratégie $\sigma : A$ est *filaire* si pour toute partie sn de σ, n est justifié par un coup de $\lceil s \rceil$, et si de plus :

si $smn \in \sigma$, si $t \in \sigma$, si $tm \in \mathcal{P}_A$ et si $\lceil sm \rceil = \lceil tm \rceil$ alors la partie tmn est dans σ (où tmn est obtenue en faisant pointer le coup n sur le coup de $\lceil tm \rceil$ qui correspond au pointeur de n dans smn).

Exercice 15
Montrer que la stratégie définie juste après l'exercice 12 n'est pas filaire.

Remarque : En terme de représentation fonctionnelle des stratégies, une stratégie filaire est obtenue en demandant dans la définition 10 qu'on ait une fonction partielle des *fils* de longueur impaire dans les coups de Joueur.

Intuitivement, cela signifie que deux appels successifs (ou entrelacés) à σ (*i.e.* morceaux de parties commençant par un coup initial) sont indépendants et que σ répondra deux fois la même chose si on l'interroge deux fois de la même manière. Il n'y a pas de mémoire d'un appel sur l'autre.

Exercice 16
Montrer qu'il y a une correspondance bijective entre la définition 18 de stratégie filaire et la représentation fonctionnelle ci-dessus.

Lemme 12
Soit s une partie dans une stratégie filaire, $\lceil s \rceil$ est une partie.

Démonstration : Il faut montrer que $\lceil s \rceil$ est alternée. On montre, par récurrence sur la longueur de s, que tous ses fils sont alternés et terminent par un coup de Joueur. Si $s = \varepsilon$, le résultat est immédiat. Sinon $s = tmn$, où n pointe dans $\lceil tm \rceil$. Par hypothèse de récurrence, tous les fils de t sont alternés et terminent par un coup de Joueur. Les fils de s sont identiques à ceux de t excepté celui qui contient m et n qui est obtenu en rajoutant un coup d'Opposant suivi d'un coup de Joueur au fil correspondant de t et donne donc un fil alterné terminant par un coup de Joueur. $\square$

Lemme 13

Soient σ une stratégie filaire sur A et $s \in \mathcal{P}_A$, $s \in \sigma$ ssi *tous les fils de s sont dans σ.*

Démonstration : On raisonne par récurrence sur la longueur de s :

- Si $s = \varepsilon$, immédiat.
- Si s est de longueur impaire, on ne peut ni avoir $s \in \sigma$, ni tous les fils de s dans σ (car ils sont disjoints).
- Si $s = tmn^{\mathsf{J}}$, si $s \in \sigma$, par hypothèse de récurrence, tous les fils de t sont dans σ, or n pointe dans $\lceil tm \rceil$ donc il suffit de montrer que $\lceil tm \rceil n = \lceil tmn \rceil \in \sigma$. Par définition d'une stratégie filaire, $\lceil tm \rceil = \lceil \lceil tm \rceil \rceil$ implique que $\lceil tm \rceil n \in \sigma$. Réciproquement, si tous les fils de s sont dans σ, on a $t \in \sigma$ par hypothèse de récurrence, $tm \in \mathcal{P}_A$, $\lceil \lceil tm \rceil \rceil = \lceil tm \rceil$ et $\lceil tm \rceil n = \lceil tmn \rceil \in \sigma$ donc $tmn \in \sigma$. $\square$

Lemme 14

id_A est une stratégie filaire sur $A \to A$.

Démonstration : Il faut tout d'abord montrer qu'un coup de Joueur pointe toujours dans son fil. On raisonne par récurrence sur la longueur de la partie s de id_A. Si s est vide, le résultat est immédiat. Si $s = tmn$, on peut décomposer s en $s = s_0\, m_0^{\mathsf{O}}\, n_0^{\mathsf{J}}\, s_1\, m^{\mathsf{O}}\, n^{\mathsf{J}}$ où n_0 justifie m (sinon m est initial et n pointe sur m), alors n pointe sur m_0 et par hypothèse de récurrence n_0 pointe dans $\lceil s_0 m_0 \rceil$ d'où $m_0 \in \lceil s_0 m_0 n_0 s_1 m \rceil$.

Soient smn et t deux parties de id_A telles que $tm \in \mathcal{P}_{A \to A}$ et $\lceil sm \rceil = \lceil tm \rceil$, on doit montrer que $tmn \in id_A$, c'est-à-dire que $tmn \upharpoonright_{A_1} = tmn \upharpoonright_{A_2}$. Supposons que $m \in A_1$, puisque $smn \upharpoonright_{A_1} = smn \upharpoonright_{A_2}$, on en déduit que $n \in A_2$ et que "$m = n$". Il reste à montrer que le pointeur de n dans $tmn \upharpoonright_{A_2}$ est le même que celui de m dans $tmn \upharpoonright_{A_1}$, ce qui se déduit de $\lceil sm \rceil = \lceil tm \rceil$. $\square$

Lemme 15

Si $\sigma : A \to B$ et $\tau : B \to C$ sont des stratégies filaires alors $\sigma;\tau : A \to C$ est une stratégie filaire.

Démonstration : On doit d'abord montrer que si $sm \in \sigma;\tau$, alors m pointe dans $\lceil s \rceil$. Soit $u \in \mathrm{int}(A, B, C)$ qui mène à sm, si $m \in C$, $m \in u \upharpoonright_{B,C}$ or les pointeurs entre coups de C sont les mêmes dans s et dans $u \upharpoonright_{B,C}$ et puisque $u \upharpoonright_{B,C} \in \tau$, m pointe dans le fil de la partie qui précède. Si $m \in A$ est un coup initial de A, il pointe sur un coup initial de B dans u et le dernier coup de s pointe héréditairement sur le même puisque $u \upharpoonright_{A,B} \in \sigma$ donc ces deux coups pointent héréditairement sur le même coup initial de C dans s. Si enfin $m \in A$ et n'est pas un coup initial, il pointe dans A tout comme le coup précédent et il pointe dans $\lceil u \upharpoonright_{A,B} \rceil$, or tout coup de A dans $\lceil u \upharpoonright_{A,B} \rceil$ est également dans $\lceil s \rceil$ ce qui permet de conclure.

Si $smn \in \sigma;\tau$, $t \in \sigma;\tau$, $t \in \mathcal{P}_{A \to C}$ et si $\lceil sm \rceil = \lceil tm \rceil$, soit $u = u'u'' \in \mathrm{int}(A, B, C)$ un témoin tel que $u \upharpoonright_{A,C} = smn$ avec $u' \upharpoonright_{A,C} = sm$ et soit $v' \in \mathrm{int}(A, B, C)$ un témoin tel que $v' \upharpoonright_{A,C} = tm$, on a $\lceil u' \upharpoonright_{A,C} \rceil = \lceil sm \rceil = \lceil tm \rceil = \lceil v' \upharpoonright_{A,C} \rceil$ donc $\lceil u' \rceil = \lceil v' \rceil$ par le lemme de zipping. On note v l'interaction dans $\mathrm{int}(A, B, C)$ obtenue en ajoutant à v' les coups de u'' qui pointent dans $\lceil u' \rceil$ (ce qui a un sens parce que $\lceil u' \rceil = \lceil v' \rceil$). On va montrer que $v \restriction_{A,C} = tmn$, $v \restriction_{A,B} \in \sigma$ et $v \restriction_{B,C} \in \tau$ et ce qui permet de conclure que $tmn \in \sigma;\tau$. Le premier résultat est direct car n est l'unique coup dans A et C de u'' et il pointe dans $\lceil sm \rceil$ d'après ce que l'on a vu ci-dessus. Pour montrer que $v \restriction_{A,B} \in \sigma$, on va prouver que tous ses fils sont dans σ (lemme 13) pour les fils inclus dans $v' \restriction_{A,B}$, le résultat est immédiat, et si on considère un fil contenant des coups provenant de u'', ce fil est égal au fil correspondant dans $u \restriction_{A,B}$ qui est lui-même dans σ. De même pour $v \restriction_{B,C} \in \tau$. □

Exercice 17
Montrer qu'une stratégie est filaire *ssi* c'est un morphisme de comonoïdes dans la catégorie $\mathcal{G}$. En déduire une preuve directe du lemme 15.

Proposition 4 (Catégorie de jeux filaires)
En restreignant la catégorie $\mathcal{G}$ aux stratégies filaires, on obtient une sous-catégorie notée $\mathcal{G}_{\text{fil}}$.

Exercice 18 (Constructions filaires)
Montrer que les projections $\pi_A : A \times B \to A$ et $\pi_B : A \times B \to B$ sont des stratégies filaires. Montrer que si σ et τ sont deux stratégies filaires, alors $\sigma \times \tau$, $\sigma \to \tau$ et $\langle \sigma, \tau \rangle$ sont filaires.

Théorème 1
La catégorie $\mathcal{G}_{\text{fil}}$ est une catégorie cartésienne fermée.

Démonstration : Puisque $\mathcal{G}_{\text{fil}}$ est une sous-catégorie de $\mathcal{G}$, en utilisant les propriétés de $\times$ et $\to$ dans $\mathcal{G}$, il suffit de montrer que $\langle \sigma, \tau \rangle$ est la seule stratégie filaire à faire commuter le diagramme du lemme 11. On montre que si $\rho : C \to A \times B$ est une stratégie filaire, alors $\rho = \langle \rho; \pi_A, \rho; \pi_B \rangle$. Il en découle que si ρ fait commuter le diagramme, $\rho; \pi_A = \sigma$ et $\rho; \pi_B = \tau$ donc $\rho = \langle \sigma, \tau \rangle$.
Soit $\rho : C \to A \times B$ une stratégie filaire, soit $s \in \langle \rho; \pi_A, \rho; \pi_B \rangle$ un fil, par l'exercice 12, $s \restriction_{C^A \to A} \in \rho; \pi_A$ et $s \restriction_{C^B \to B} \in \rho; \pi_B$, or $s \restriction_{C^A \to A} = s$ ou $s \restriction_{C^A \to A} = \varepsilon$ (car s est un fil dans $C \to A \times B$) et de même pour $s \restriction_{C^B \to B}$ donc $s \in \rho$. Réciproquement, si $s \in \rho$ est un fil, $s \restriction_{C^A \to A} = \varepsilon$ ou $s \restriction_{C^A \to A} = s$ donc $s \restriction_{C^A \to A} \in \rho$, et on voit facilement que $s \restriction_{C^A \to A} \in \rho; \pi_A$ car elle ne contient pas de coup dans B. □

Remarque : On vient de montrer techniquement le fait que le calcul effectué par une stratégie filaire est indépendant d'un appel à l'autre. Deux appels d'une stratégie ρ filaire sur $C \to A \times B$ portant l'un sur A et l'autre sur B ne peuvent pas communiquer, ce qui impose l'indépendance des deux composantes et donc le fait que ρ puisse être décomposée en deux stratégies indépendantes dans $C \to A$ et $C \to B$.

Exercice 19 (Monade de double décalage)
Montrer que $(\_ \to \bot) \to \bot$ est une monade sur $\mathcal{G}_{\text{fil}}$. Décrire la catégorie de Kleisli de $\mathcal{G}_{\text{fil}}$ par rapport à cette monade.

1.4 Interprétation du λ-calcul

D'après le théorème 1, les jeux filaires donnent un modèle dénotationnel du λ-calcul. Pour clarifier les idées, nous allons redonner directement l'interprétation du λ-calcul dans les jeux.

On utilise le système de typage suivant :

$$\frac{}{x : A \vdash x : A}\ \text{var}$$

$$\frac{\Gamma \vdash u : B}{\Gamma \setminus \{x : A\} \vdash \lambda x.u : A \to B}\ \text{lam}$$

$$\frac{\Gamma \vdash u : A \to B \qquad \Gamma' \vdash v : A}{\Gamma \cup \Gamma' \vdash (u)v : B}\ \text{app}$$

Un type A est interprété par l'arène correspondante (du moment que l'on a fixé une interprétation arbitraire des atomes par des arènes) et une dérivation de typage de conclusion $x_1 : A_1, \dots, x_n : A_n \vdash t : A$ (ou $\Gamma \vdash t : A$) est interprétée par une stratégie σ_t sur l'arène $A_1 \times \cdots \times A_n \to A$ (ou $\Gamma \to A$). Cette interprétation est définie par induction sur la dérivation :

(var) La stratégie σ_x sur $A \to A$ obtenue est id_A.

(lam) Soit σ_u la stratégie obtenue sur $\Gamma \to B$, si A n'apparaît pas dans Γ on commence par remplacer σ_u par $\sigma_u \times \varepsilon_A$ ce qui donne dans tous les cas une stratégie sur $\Gamma' \times A \to B$. Par curryfication $\Gamma' \times A \to B = \Gamma' \to (A \to B)$, cette stratégie est également une stratégie $\Gamma' \to (A \to B)$ qui est $\sigma_{\lambda x.u}$.

(app) Soient σ_u la stratégie obtenue sur $\Gamma \to (A \to B)$ et σ_v celle sur $\Gamma' \to A$, pour pouvoir composer σ_u et σ_v, il faut remplacer σ_v par $id_\Gamma \times \sigma_v : \Gamma \times \Gamma' \to \Gamma \times A$ et on peut obtenir $(id_\Gamma \times \sigma_v); \sigma_u : \Gamma \times \Gamma' \to B$ (pour peu que l'on voie σ_u comme stratégie sur $\Gamma \times A \to B$). En précomposant par les Δ_C nécessaires pour identifier les variables communes à Γ et Γ', on obtient finalement $\sigma_{(u)v} : \Gamma \cup \Gamma' \to B$.

Cette interprétation réalise la $\beta\eta$-équivalence :

- pour montrer que $\sigma_{(\lambda x.u)v} = \sigma_{u[^v/_x]}$, il suffit de voir (par induction sur u) que $\sigma_{u[^v/_x]}$ s'obtient en précomposant σ_u par σ_v de la manière adéquate ;
- pour montrer que $\sigma_{\lambda x.(u)x} = \sigma_u$ si $x \notin u$, il suffit d'utiliser que id_A est neutre à gauche pour la composition.

Exercice 20
Montrer que l'interprétation du λ-calcul dans la catégorie $\mathcal{G}_{\text{fil}}$ peut être étendue pour donner un modèle du λ-calcul avec booléens que l'on a utilisé à la section 1.1.

Puisque la catégorie $\mathcal{G}_{\text{fil}}$ est un modèle du λ-calcul, l'interprétation d'un terme est donnée par celle de sa forme normale η-longue. On va chercher à décrire plus concrètement l'interprétation de ces formes normales.

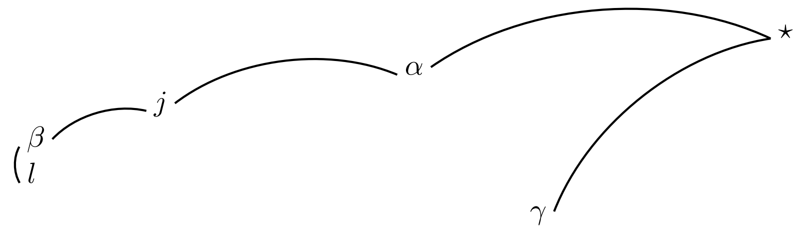
Les formes normales du λ-calcul sont définissables par la grammaire $u ::= \lambda\vec{x}.(y)\vec{u}$ (où, pour un terme clos, le premier y est dans les $\vec{x}$). Un type simple peut toujours s'écrire $A = A_1 \to \cdots \to A_n \to X$ où les A_i sont des types simples et X est un atome. Soit t un λ-terme clos de type A en forme normale η-longue, on peut décomposer t de la manière suivante :

$$\begin{aligned}
t &= \lambda\vec{x}^{A_i}.(\alpha)\vec{u}^{B_j} : A_1 \to \cdots \to A_n \to X && \text{où } \alpha \in \vec{x} \text{ et } A_i = B_1 \to \ldots \to B_m \to X\\
&= \lambda\vec{x}^{A_i}.(\alpha)u_1 \ldots (\lambda\vec{y}^{C_k}.(\beta)\vec{v}^{D_l}) \ldots u_m && \text{où } \beta \in \vec{x} \cup \vec{y} \text{ et } B_j = C_1 \to \ldots \to C_r \to Y\\
v_l &= \lambda\vec{z}^{E_p}.(\gamma)\vec{w}^{F_q} && \text{où } \gamma \in \vec{x} \cup \vec{y} \cup \vec{z} \text{ et } D_l = E_1 \to \ldots \to E_s \to Z\\
&\ldots
\end{aligned}$$

Les entiers n, m, r, s ... sont entièrement déterminés par le type. À α-équivalence près, t ne dépend que de α, β, γ, ... on va voir que c'est exactement l'information que code la stratégie qui interprète t. Une partie jouée par cette stratégie est de la forme :

$$
\begin{array}{l}
A_1^+ \rightarrow \cdots \qquad\qquad\qquad A_i^+ \qquad\qquad\qquad \cdots \rightarrow \cdots \rightarrow A_n^+ \rightarrow X^- \\
\qquad B_1^- \rightarrow \quad \cdots \quad B_j^- \quad \cdots \quad \rightarrow B_m^- \rightarrow X^+ \\
\qquad\qquad C_1^+ \rightarrow \cdots \rightarrow C_r^+ \rightarrow Y^-
\end{array}
$$

O
J
O
J
O
J



Ce qui donne :

- O demande le résultat
- J répond que c'est celui de la variable de tête α, ce qui revient à choisir le A_i correspondant (car $\alpha \in \vec{x}$)
- O demande la valeur du j^{e} argument de α, ce qui revient à donner un sous-type B_j de A_i
- J répond que la variable de tête de cet argument est β, ce qui revient à choisir l'un des C_k (si $\beta \in \vec{y}$) ou l'un des A_i (si $\beta \in \vec{x}$)
- O demande la valeur du l^{e} argument de β, ce qui revient à donner un sous-type D_l de C_k
- J répond que la variable de tête de cet argument est γ, ce qui revient à choisir l'un des E_p (si $\gamma \in \vec{z}$), l'un des C_k (si $\gamma \in \vec{y}$) ou l'un des A_i (si $\gamma \in \vec{x}$)
- ...

la partie continue selon des identités puisque c'est ainsi que l'on interprète les variables.

Exemple 21
Pour le λ-terme $\lambda f.\lambda x.(f)(f)x : (X \rightarrow X) \rightarrow X \rightarrow X$, on retrouve des parties similaires à celles de la deuxième partie des exemples 6 et 10.

Remarque : On peut relier cette présentation de la stratégie associée à un terme à la notion de *réduction linéaire de tête* du λ-calcul ainsi qu'à celle d'*arbre de Böhm*.

Exercice 21
Montrer que la description que l'on a donnée ci-dessus pour les formes normales correspond bien à la définition plus formelle d'interprétation du λ-calcul donnée précédemment.

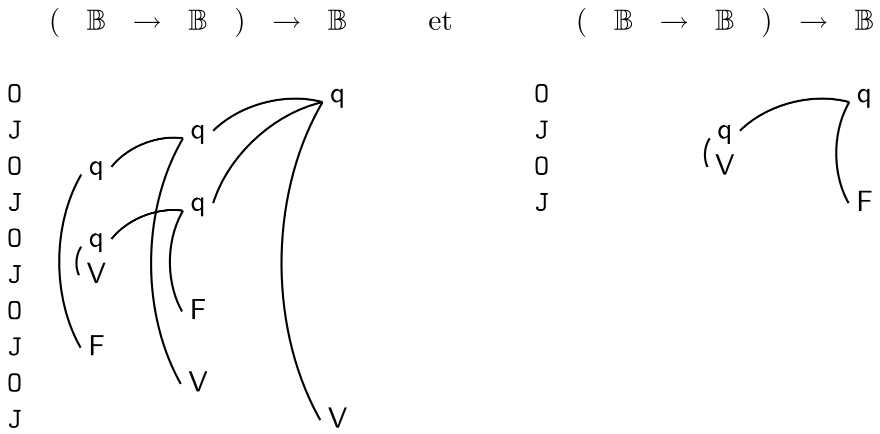
Exercice 22
Calculer l'interprétation des deux termes de Kierstead que l'on a rencontrés à la section 1.1 : $\lambda f.(f)\lambda x.(f)\lambda y.x$ et $\lambda f.(f)\lambda x.(f)\lambda y.y$, grâce à la description directe que l'on a donnée ici.

Remarque : Étant donné un modèle catégorique d'un langage, donc une interprétation des types dans les objets et des termes dans les morphismes, on dit que le modèle est *pleinement complet* (ou complet) si tout morphisme entre deux objets, qui sont l'interprétation de types, est l'interprétation d'un terme (*i.e.* si l'interprétation des termes est surjective parmi les morphismes entre interprétations de types). La terminologie vient du fait que cela correspond à dire que le foncteur interprétation est *plein*, ou encore que la sous-catégorie formée des interprétations de termes est une *sous-catégorie pleine* du modèle.

À partir de tout modèle, il est facile de construire un modèle complet en se restreignant aux interprétations de termes, mais ce résultat n'a aucun intérêt ! On cherche à trouver des caractérisations sémantiques (indépendantes de la syntaxe) de ces interprétations de termes.

L'interprétation du λ-calcul dans $\mathcal{G}_{\text{fil}}$ n'est pas complète comme le montre l'exemple ci-dessous. On va essayer de construire pourtant des modèles de jeux complets, soit en augmentant le langage, soit en restreignant encore le modèle (donc les stratégies).

Exemple 22
La stratégie contenant les deux parties suivantes peut être définie de manière filaire :



cependant, elle ne correspond au comportement d'aucun λ-terme puisqu'elle est capable de définir sa réponse finale en fonction de la manière dont l'argument a calculé V. On peut comparer à l'interprétation de $\lambda f.(f)(f)\mathsf{V}$.

1.5 Innocence

Définition 19 (Vue)
Soit s une partie de l'arène A, la *vue* $\ulcorner s \urcorner$ de s est la sous-suite pointée de s définie récursivement par :
- $\ulcorner \varepsilon \urcorner = \varepsilon$;
- $\ulcorner s n^{\mathsf{J}} \urcorner = \ulcorner s \urcorner n^{\mathsf{J}}$;
- $\ulcorner s n^{\mathsf{O}} \urcorner = n^{\mathsf{O}}$ si n est initial ;
- $\ulcorner s m t n^{\mathsf{O}} \urcorner = \ulcorner s \urcorner m n = \ulcorner s m \urcorner n$ si m justifie n dans $smtn$.

C'est une partie sur A.

Remarque : Tout coup d'Opposant de la vue d'une partie s pointe sur le coup de Joueur situé juste avant. On appellera *vue* toute partie dans laquelle tous les coups d'Opposant pointent sur le coup situé juste avant. Un partie s est une vue *ssi* $s = \ulcorner s \urcorner$.

Remarque : L'interprétation des formes normales η-longues donnée à la section 1.4 est définie en terme de vues.

Lemme 16
Soit s une partie sur A telle que si $tn^J \leq s$, n pointe dans $\ulcorner t \urcorner$, alors on a $\ulcorner s \urcorner \subset \lceil s \rceil$.

Démonstration : On va montrer que tous les coups de $\ulcorner s \urcorner$ sont héréditairement justifiés par le même coup initial. On raisonne par induction sur la définition de $\ulcorner s \urcorner$:

– si $s = \varepsilon$, immédiat ;
– si $s = tn^{\mathsf{J}}$, par hypothèse, n pointe dans $\ulcorner t \urcorner$ et par hypothèse de récurrence, tous les coups de $\ulcorner t \urcorner$ pointent héréditairement sur le même coup initial donc c'est le cas pour $\ulcorner s \urcorner = \ulcorner t \urcorner n$;
– si $s = tn^{\mathsf{O}}$, avec n^{O} initial, immédiat ;
– si $s = s'mtn^{\mathsf{O}}$ et n pointe sur m, alors par hypothèse de récurrence, tous les coups de $\ulcorner s'm \urcorner$ pointent héréditairement sur le même coup initial or $\ulcorner s'mtn \urcorner = \ulcorner s'm \urcorner n$ et n pointe héréditairement sur le même coup initial que m. □

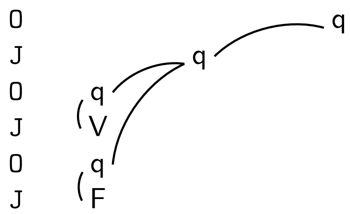
Définition 20 (Stratégie innocente)
Une stratégie $\sigma : A$ est *innocente* si pour toute partie sn de σ, n est justifié par un coup de $\ulcorner s \urcorner$ (condition de *visibilité*), et si de plus :

si $smn \in \sigma$, si $t \in \sigma$, si $tm \in \mathcal{P}_A$ et si $\ulcorner sm \urcorner = \ulcorner tm \urcorner$ alors la partie tmn est dans σ (où tmn est obtenue en faisant pointer le coup n sur le coup de $\ulcorner tm \urcorner$ qui correspond au pointeur de n dans smn).

Remarque : En terme de représentation fonctionnelle des stratégies, une stratégie innocente est obtenue en demandant dans la définition 10 qu'on ait une fonction partielle des *vues* de longueur impaire dans les coups de Joueur. On parle alors de *fonction de vue*.

Exemple 23
La stratégie contenant la trace suivante n'est pas innocente :

$$(\ \mathbb{B} \ \rightarrow\ \mathbb{B}\ )\ \rightarrow\ \mathbb{B}$$

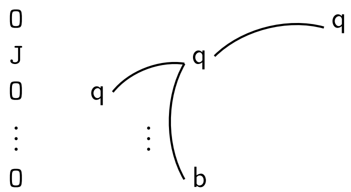


elle utilise plus que sa vue pour déterminer sa réponse.

Exercice 23
Montrer que la stratégie donnée à l'exemple 22 n'est pas innocente.

Remarque : Intuitivement, une stratégie innocente est une stratégie qui ne dépend pas des calculs intermédiaires d'Opposant (pour un même résultat), qui sont les morceaux cachés par la vue. Les vues commencent à être non triviales à partir du type $(\mathbb{B} \rightarrow \mathbb{B}) \rightarrow \mathbb{B}$. Dans une partie de la forme :

$$(\ \mathbb{B} \ \rightarrow\ \mathbb{B}\ )\ \rightarrow\ \mathbb{B}$$



la réponse finale de Joueur ne peut dépendre que de la valeur de b car, quoi qu'il se soit passé entre le deuxième et le dernier coup, c'est-à-dire quelle que soit la manière dont Opposant a calculé son résultat b, la vue est la même.

C'est ce que montre l'exemple 22, une stratégie innocente n'a pas le droit de profiter des informations issues de la façon dont Opposant calcule ses résultats.

Exercice 24
Montrer (en ajoutant les pointeurs) que l'exemple 7 donne une stratégie non innocente. Comment corriger cet exemple pour obtenir un "goûteur de et" innocent ?

Lemme 17
Toute stratégie innocente est filaire.

DÉMONSTRATION : Conséquence directe du lemme 16. □

Lemme 18
id_A est une stratégie innocente sur $A \to A$.

DÉMONSTRATION : Essentiellement la même preuve que pour le lemme 14. □

Lemme 19
Si $\sigma : A \to B$ et $\tau : B \to C$ sont des stratégies innocentes alors $\sigma;\tau : A \to C$ est une stratégie innocente.

Proposition 5 (Catégorie de jeux innocents)
En restreignant la catégorie $\mathcal{G}_{\mathrm{fil}}$ aux stratégies innocentes, on obtient une sous-catégorie notée $\mathcal{G}_{\mathrm{inn}}$.

Théorème 2
La catégorie $\mathcal{G}_{\mathrm{inn}}$ est une catégorie cartésienne fermée.

Remarque : L'interprétation du λ-calcul dans $\mathcal{G}_{\mathrm{inn}}$ est la même que celle dans $\mathcal{G}_{\mathrm{fil}}$, puisque les variables sont interprétées par des stratégies *id* qui sont innocentes et puisque toutes les constructions utilisées préservent l'innocence. On constate aussi simplement que l'interprétation directe d'un λ-terme en forme normale η-longue que l'on a décrite est non seulement filaire mais en fait innocente.

1.6 Incomplétude

Définition 21 (Stratégie innocente finie)
Une stratégie innocente est *finie* si le graphe de sa fonction de vue est un graphe fini. On note alors $|\sigma|$ la somme des longueurs des vues du graphe.

Remarque : On va enfin pouvoir décrire des stratégies in extenso, ce qui est nettement plus difficile pour des stratégies qui ne sont pas innocentes finies.

Définition 22 (Stratégie totale)
Une stratégie est *totale* si sa représentation fonctionnelle est une fonction totale.

Remarque : L'interprétation d'un λ-terme dans $\mathcal{G}_{\mathrm{inn}}$ est une stratégie finie totale.

Exemple 24
La stratégie sur $(\perp \to \perp) \to \perp$ définie par les préfixes de la partie suivante :

$$\begin{array}{cccccccc}
 & (& \perp & \to & \perp &) & \to & \perp \\
\mathsf{O} & & & & & & & \star \\
\mathsf{J} & & & & \star & & & \\
\mathsf{O} & & \star & & & & & \\
\mathsf{J} & & & & \star & & & \\
\mathsf{O} & & \star & & & & & \\
\mathsf{J} & & & & \star & & & \\
\mathsf{O} & & \star & & & & & \\
 & & & \vdots & & & &
\end{array}$$

est une stratégie innocente totale mais pas finie.

Il existe une stratégie innocente finie totale qui n'est l'interprétation d'aucun λ-terme.

$$\begin{array}{cccccccc}
 & (& \mathbb{B} & \to & \mathbb{B} &) & \to & \mathbb{B} \\
\mathsf{O} & & & & & & & \mathsf{q} \\
\mathsf{J} & & \mathsf{q} & & & & & \\
\mathsf{O} & & \mathsf{b} & & & & & \\
\mathsf{J} & & & & & & & \mathsf{F}
\end{array}
\qquad \text{et} \qquad
\begin{array}{cccccccc}
 & (& \mathbb{B} & \to & \mathbb{B} &) & \to & \mathbb{B} \\
\mathsf{O} & & & & & & & \mathsf{q} \\
\mathsf{J} & & & & \mathsf{q} & & & \\
\mathsf{O} & & \mathsf{q} & & & & & \\
\mathsf{J} & & & & & & & \mathsf{V}
\end{array}$$

Une vue de longueur 3 sur le type $(\mathbb{B}_2 \to \mathbb{B}_1) \to \mathbb{B}_0$ commence par le coup q_0 d'Opposant, suivi par le coup q_1 de Joueur, il y a alors deux types de coups possibles d'Opposant pointant sur q_1 : V_1 et F_1 ou q_2. Si Joueur répond alors un coup dans $\mathbb{B}_0$, Opposant ne peut plus continuer en jouant une vue (ou alors on revient à une vue plus courte déjà décrite). Les parties ci-dessus donnent donc intégralement une stratégie innocente totale finie. Cependant, une partie jouée par un λ-terme dans $(\mathbb{B} \to \mathbb{B}) \to \mathbb{B}$ doit répondre aux questions dans l'ordre inverse de celui dans lequel elles ont été posées, ce qui montre que la partie de droite n'est pas jouable par un λ-terme.

Attention à ne pas confondre ce qui se passe ici avec le cas de l'exemple 22 qui correspond à une stratégie filaire non innocente.

On va donner deux réponses au fait que $\mathcal{G}_{\text{inn}}$ ne fournit pas un modèle complet du λ-calcul. La première consiste à étendre le langage et à montrer que $\mathcal{G}_{\text{inn}}$ est complète pour le $\lambda\mu$-calcul (section 2.4). La seconde nécessite de rajouter encore une contrainte sur les stratégies et donne un résultat de complétude des stratégies *bien parenthésées* pour le λ-calcul (section 3.3).

2 Modèle de jeux du $\lambda\mu$-calcul

La section précédente a consisté à construire un modèle du λ-calcul mais nous avons vu que certaines stratégies ne sont pas l'interprétation de λ-termes. Nous allons introduire le $\lambda\mu$-calcul qui est une extension du λ-calcul. Nous montrerons que le modèle de jeux que nous avons construit est également un modèle du $\lambda\mu$-calcul et que l'on aboutit à un résultat de complétude.

2.1 $\lambda\mu$-calcul

À travers la correspondance de Curry-Howard, le λ-calcul simplement typé correspond à la logique intuitionniste minimale. De nombreux travaux dans les années 90 ont permis d'étendre cette correspondance à la logique classique en montrant que les instructions de contrôle (call/cc, exceptions, jump, ...) peuvent être typées par des formules vraies en logique classique et pas en logique intuitionniste (loi de Peirce $((A \to B) \to A) \to A$, tiers exclus $A \vee \neg A$, raisonnement par l'absurde $\neg\neg A \to A$, ...).

Nous allons nous concentrer sur un calcul particulier qui se présente comme une extension du λ-calcul : le $\lambda\mu$-calcul [17] de M. Parigot.

Les termes du $\lambda\mu$-calcul sont construits à partir de deux ensembles de variables : les λ-variables x, y, ... et les μ-variables α, β ... et contiennent les λ-termes :

$$t \quad ::= \quad x \quad | \quad \lambda x.t \quad | \quad (t)t \quad | \quad \mu\alpha[\beta]t$$

Cette syntaxe contient deux lieurs : λ pour les λ-variables et μ pour les μ-variables. Dans la construction $\mu\alpha[\beta]t$, β est libre (sauf si $\beta = \alpha$).

Si t est un $\lambda\mu$-terme, $n = [\beta]t$ est appelé un *terme nommé*. Si n est un terme nommé, $\mu\alpha.n$ est un $\lambda\mu$-terme.

Pour le système des types simples, on utilise les mêmes types que pour le λ-calcul et les jugements sont de la forme : $x_1 : A_1, ..., x_n : A_n \vdash t : A \mid \alpha_1 : B_1, ..., \alpha_k : B_k$. Les règles de typage étendent celles du λ-calcul :

$$\frac{}{x : A \vdash x : A \mid}\ \text{var}$$

$$\frac{\Gamma \vdash u : B \mid \Delta}{\Gamma \setminus \{x : A\} \vdash \lambda x.u : A \to B \mid \Delta}\ \text{lam} \qquad \frac{\Gamma \vdash u : A \to B \mid \Delta \qquad \Gamma' \vdash v : A \mid \Delta'}{\Gamma \cup \Gamma' \vdash (u)v : B \mid \Delta \cup \Delta'}\ \text{app}$$

$$\frac{\Gamma \vdash u : B \mid \Delta}{\Gamma \vdash \mu\alpha[\beta]u : A \mid (\Delta \cup \{\beta : B\}) \setminus \{\alpha : A\}}\ \text{mu}$$

Exemple 25 (Loi de Peirce)
Le terme $\lambda f.\mu\alpha[\alpha](f)\lambda x.\mu\delta[\alpha]x$ est typable de type $((A \to B) \to A) \to A$.

Les règles de réduction complètent celles du λ-calcul avec de nouvelles étapes associées au lieur μ :

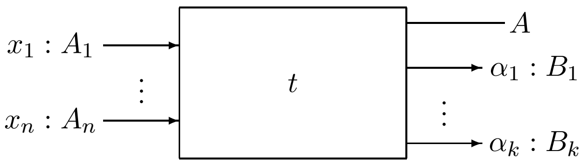
$$\begin{array}{rll} (\mu\alpha.n)u & \to_\mu & \mu\alpha.n[{}^{[\alpha](v)u}/_{[\alpha]v}] \\ [\beta]\mu\alpha.n & \to_\rho & n[{}^{\beta}/_{\alpha}] \\ \mu\alpha[\alpha]t & \to_\theta & t \qquad\qquad\qquad\qquad \alpha \notin t \end{array}$$

La μ-substitution $t[{}^{[\alpha](v)u}/_{[\alpha]v}]$ est définie par récurrence sur t :

$$\begin{aligned} x[{}^{[\alpha](v)u}/_{[\alpha]v}] &= x \\ (\lambda x.t)[{}^{[\alpha](v)u}/_{[\alpha]v}] &= \lambda x.(t[{}^{[\alpha](v)u}/_{[\alpha]v}]) \\ ((t)t')[{}^{[\alpha](v)u}/_{[\alpha]v}] &= (t[{}^{[\alpha](v)u}/_{[\alpha]v}])t'[{}^{[\alpha](v)u}/_{[\alpha]v}] \\ (\mu\gamma[\beta]t)[{}^{[\alpha](v)u}/_{[\alpha]v}] &= \mu\gamma[\beta](t[{}^{[\alpha](v)u}/_{[\alpha]v}]) \qquad\qquad \beta \neq \alpha \\ (\mu\gamma[\alpha]t)[{}^{[\alpha](v)u}/_{[\alpha]v}] &= \mu\gamma[\alpha](t[{}^{[\alpha](v)u}/_{[\alpha]v}])u \end{aligned}$$

2.2 Interprétation calculatoire

On peut comprendre un $\lambda\mu$-terme comme un programme à plusieurs sorties nommées par des μ-variables. Si $x_1 : A_1, ..., x_n : A_n \vdash t : A \mid \alpha_1 : B_1, ..., \alpha_k : B_k$, on peut voir t comme :



Dans un β-rédex $(\lambda x.t)u$, la variable x précise à quelle entrée du terme t l'argument u doit être passé. Le μ-rédex $(\mu\alpha.t)u$ introduit un autre mécanisme de routage qui précise d'abord vers quel canal (ou sortie) l'argument va être routé pour se retrouver ensuite en position de β-rédex. Ce mécanisme n'apparaît pas dans le λ-calcul puisque les termes n'ont qu'une sortie.

Nous allons montrer comment il est possible d'interpréter un calcul d'exceptions dans le $\lambda\mu$-calcul. On utilise pour cela la machine de Krivine [10] (dans laquelle on a omis certains environnements) :

$(\lambda f.\mu\alpha[\alpha](f)\lambda x.\mu\delta[\alpha]x)\lambda k.t$		π	*push*
$\lambda f.\mu\alpha[\alpha](f)\lambda x.\mu\delta[\alpha]x$		$\lambda k.t :: \pi$	*pop*
$\mu\alpha[\alpha](f)\lambda x.\mu\delta[\alpha]x$	$(f = \lambda k.t)$	π	*save*
$[\alpha](f)\lambda x.\mu\delta[\alpha]x$	$(f = \lambda k.t) + (\alpha = \pi)$		*restore*
$(f)\lambda x.\mu\delta[\alpha]x$	$(f = \lambda k.t) + (\alpha = \pi)$	π	*push*
f	$(f = \lambda k.t) + (\alpha = \pi)$	$\lambda x.\mu\delta[\alpha]x :: \pi$	*deref*
$\lambda k.t$	$(f = \lambda k.t) + (\alpha = \pi)$	$\lambda x.\mu\delta[\alpha]x :: \pi$	*pop*
t	$(f = \lambda k.t) + (\alpha = \pi) + (k = \lambda x.\mu\delta[\alpha]x)$	π	

Un sous-terme de t de la forme $(k)u$ correspond à un `raise` k avec la continuation u.

Pour plus de détails sur ces inteprétations calculatoires, on pourra se référer à [15, 14].

2.3 Modèle de jeux

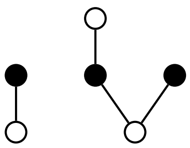
Nous allons construire une interprétation du $\lambda\mu$-calcul dans les jeux qui correspond au modèle décrit par J. Laird [11].

Afin d'interpréter les jugements de typage avec plusieurs formules à droite, il est nécessaire de définir une nouvelle construction d'arènes.

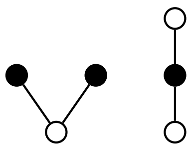
Définition 23 (Arène somme)
Si A et B sont deux arènes, l'arène somme $A + B$ est obtenue en construisant un arbre par identification des racines pour chaque paire d'un arbre dans A et d'un arbre de B.

On dira qu'un coup de $A + B$ est dans A (resp. B) si c'est une racine ou si c'est un coup qui provient d'un arbre de A (resp. B).

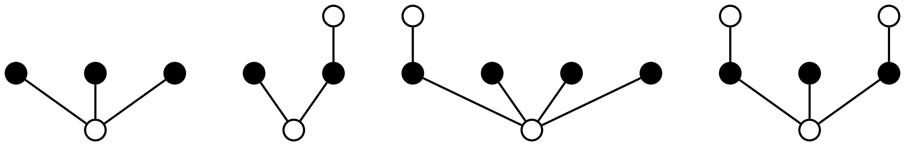
Exemple 26
À partir des arènes A et B suivantes :



et



on obtient l'arène $A + B$:



Exercice 25
Montrer que, si A et B sont deux arènes, $A + \bot = A$ et $A \to B = (A \to \bot) + B$

Définition 24 (Affaiblissement)
Soit A une arène, la stratégie aff_A sur l'arène $\bot \to A$ est $\{a\circ \mid a \in A \wedge \circ \in \bot\}$.

Définition 25 (Contraction)
Soit A une arène, on définit la stratégie ctr_A sur l'arène $A + A \to A$. Si on note cette arène $A_1 + A_2 \to A_0$ pour distinguer les occurrences de A, pour toute partie s sur cette arène, on note s_i ($i = 1$ ou 2) la sous-suite pointée de s contenant les coups initiaux de A_0, les coups de A_i et ceux de A_0 tels que le dernier coup situé avant dans $A_1 + A_2$ est dans A_i. On a alors $\mathit{ctr}_A = \{s \in \mathcal{P}_{A_1+A_2\to A_0} \mid s_1 \in \mathit{id}_A \wedge s_2 \in \mathit{id}_A\}$.

Exemple 27 (Contraction sur $\mathbb{B}$)
La contraction pour l'arène $\mathbb{B}$ est la stratégie sur $\mathbb{B} + \mathbb{B} \to \mathbb{B}$ engendrée par les vues suivantes :

	$\mathbb{B}$	$+$	$\mathbb{B}$	$\to$	$\mathbb{B}$
O					q
J			q		
O	b				
J					b

et

	$\mathbb{B}$	$+$	$\mathbb{B}$	$\to$	$\mathbb{B}$
O					q
J			q		
O			b		
J					b

Exercice 26
Montrer que aff_A et ctr_A sont des stratégies innocentes totales finies.

On interprète une dérivation de typage terminant par le jugement $x_1 : A_1, ..., x_n : A_n \vdash t : A \mid \alpha_1 : B_1, ..., \alpha_k : B_k$ (resp. $\Gamma \vdash A \mid \Delta$) par une stratégie sur l'arène $A_1 \times \cdots \times A_n \to A + B_1 + \cdots + B_k$ (resp. $\Gamma \to A + \Delta$).

Il suffit de donner l'interprétation de la règle (mu). Si σ_u est une stratégie sur $\Gamma \to B + \Delta$, on considère quatre cas :

- Si $\alpha : A \in \Delta$ et $\beta : B \in \Delta$, on peut écrire $\sigma_u : \Gamma \to B + B + A + \Delta'$, en composant σ_u avec ctr_B, on obtient $\sigma_{\mu\alpha[\beta]u} : \Gamma \to A + B + \Delta'$.
- Si $\alpha : A \in \Delta$ et $\beta : B \notin \Delta$, on peut écrire $\sigma_u : \Gamma \to B + A + \Delta'$, on a directement $\sigma_{\mu\alpha[\beta]u} = \sigma_u : \Gamma \to A + B + \Delta'$.
- Si $\alpha : A \notin \Delta$ et $\beta : B \in \Delta$, on peut écrire $\sigma_u : \Gamma \to B + B + \Delta'$, en composant σ_u avec aff_A et ctr_B, on obtient $\sigma_{\mu\alpha[\beta]u} : \Gamma \to A + B + \Delta'$.
- Si $\alpha : A \notin \Delta$ et $\beta : B \notin \Delta$, on peut écrire $\sigma_u : \Gamma \to B + \Delta'$, en composant σ_u avec aff_A, on obtient $\sigma_{\mu\alpha[\beta]u} : \Gamma \to A + B + \Delta'$.

Exercice 27 (Somme de stratégies)
Pour définir l'interprétation précédente, on a implicitement utiliser la stratégie $\mathit{id}_A + \sigma : A + B \to A + C$ définie à partir de id_A et $\sigma : B \to C$. Définir cette stratégie pour toute stratégie σ. Est-il possible, de manière générale, de définir $\tau + \sigma$?

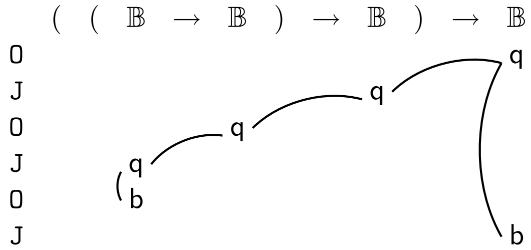
Cette interprétation réalise les μ, ρ et θ réductions :

(μ) En utilisant $\Gamma \to (A \to B) + \Delta = \Gamma \to A \to (B + \Delta) = (\Gamma \times A) \to (B + \Delta)$, on montre par récurrence sur n que la stratégie $\sigma_{\mu\alpha.n[{}^{[\alpha](v)u}/_{[\alpha]v}]}$ s'obtient en composant $\sigma_{\mu\alpha.n}$ avec σ_u tout comme $\sigma_{(\mu\alpha.n)u}$.

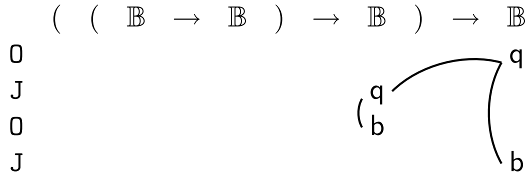
(ρ) Si $\alpha \notin n$, on a immédiatement $\sigma_{[\beta]\mu\alpha.n} = \sigma_{n[{}^{\beta}/_{\alpha}]}$. Sinon, soit $\beta \in n$, et $\sigma_{[\beta]\mu\alpha.n}$ s'obtient en composant σ_n avec une contraction (tout comme $\sigma_{n[{}^{\beta}/_{\alpha}]}$), soit $\beta \notin n$ et $\sigma_{[\beta]\mu\alpha.n} = \sigma_n = \sigma_{n[{}^{\beta}/_{\alpha}]}$.

(θ) Par définition de l'interprétation de $\mu\alpha[\beta]t$, on a $\sigma_{\mu\alpha[\alpha]t} = \sigma_t$ si $\alpha \notin t$.

Exemple 28 (call/cc)
L'interprétation du terme $\lambda f.\mu\alpha[\alpha](f)\lambda x.\mu\delta[\alpha]x$ de type $((\mathbb{B} \to \mathbb{B}) \to \mathbb{B}) \to \mathbb{B}$ est la stratégie contenant les vues suivantes et leurs préfixes de longueur paire :



et



Exemple 29 (catch)
La stratégie décrite page 29 après l'exemple 24 n'est l'interprétation d'aucun λ-terme mais c'est l'interprétation du terme $\lambda f.\mu\alpha[\alpha]$ if $(f)\mu\delta[\alpha]$V then F else F.

Remarque : Les modèles catégoriques du λ-calcul sont réalisés par les catégories cartésiennes fermées, on peut étendre ces résultats pour le $\lambda\mu$-calcul ce qui donne lieu à la notion de *catégorie de contrôle* [19]. On pourrait en particulier montrer que notre modèle de jeu constitue une catégorie de contrôle [13].

Comme on l'a fait pour le λ-calcul, il est possible de donner une description directe de la stratégie qui interprète une forme normale η-longue.

2.4 Complétude

On peut établir notre premier résultat de complétude : "toute stratégie est l'interprétation d'un terme".

Afin de simplifier le plus possible la situation, on considère le $\lambda\mu$-calcul simplement typé avec des types construits sur un unique type atomique $\mathbb{U}$ interprété par l'arène $\bot \to \bot$ et on ajoute aux termes une constante c et une construction de séquencement $u;v$ avec les règles de typage :

$$\frac{}{\vdash \mathsf{c} : \mathbb{U} \mid}\,\mathrm{c} \qquad \frac{\Gamma \vdash u : \mathbb{U} \mid \Delta \qquad \Gamma' \vdash v : A \mid \Delta'}{\Gamma \cup \Gamma' \vdash u;v : A \mid \Delta \cup \Delta'}\,\mathrm{seq}$$

La nouvelle constante c est interprétée par l'unique stratégie totale sur $\mathbb{U}$:

$$\begin{array}{cc} & \mathbb{U} \\ \mathtt{O} & \mathtt{q} \\ \mathtt{J} & \checkmark \end{array}$$

et pour la construction $-;-$, si $\tau : A$ est une stratégie, on lui associe $\uparrow\tau$ sur $\mathbb{U} \to A$ définie par $\{\varepsilon\} \cup \{aq \mid a \in A\} \cup \{aq\checkmark s \mid as \in \tau\}$. L'interprétation de $u;v$ est obtenue en composant σ_u et $\uparrow\sigma_v$.

Remarque : Si on étendait ce langage avec la possibilité que les programmes ne terminent pas, on obtiendrait un langage appelé PCF unaire.

Théorème 3 (Complétude pour le $\lambda\mu$-calcul)
Soit A un type simple contenant comme unique atome $\mathbb{U}$, si σ est une stratégie innocente finie totale sur l'arène A, il existe un $\lambda\mu$-terme de type A dont σ est l'interprétation.

Démonstration : On se place dans le cas légèrement plus général d'une stratégie σ sur l'arène $A + \mathbb{U}_1 + \cdots + \mathbb{U}_p$ (où les indices servent à différencier les occurrences de $\mathbb{U}$) et on va montrer que σ est définissable par un terme qui peut contenir des μ-variables libres α_1, ..., α_p.
On raisonne par récurrence sur $|\sigma|$. On note $A = A_1 \to \cdots \to A_n \to \mathbb{U}$ (avec éventuellement $n = 0$ pour $A = \mathbb{U}$), q le coup initial de A et q^0_k et $\checkmark^0_k$ les coups de $\mathbb{U}_k$. Par déterminisme, σ a une unique réponse à q.

- Si c'est $\checkmark$ dans A, σ est l'interprétation du terme $\lambda x_1 \ldots \lambda x_n.\mathsf{c}$.
- Si c'est $\checkmark^0_k$, σ est l'interprétation du terme $\lambda x_1 \ldots \lambda x_n.\mu\delta[\alpha_k]\mathsf{c}$ (ou même $\mu\delta[\alpha_k]\mathsf{c}$).
- Sinon, c'est le coup initial d'un des A_i que l'on note q_i. Si $A_i = \mathbb{U}$, la réponse d'Opposant ne peut être que $\checkmark_i$ dans A_i, soit τ la stratégie définie par $\tau = \{\varepsilon\} \cup \{qs \mid qq_i\checkmark_i s \in \sigma\}$: $A + \mathbb{U}_1 + \cdots + \mathbb{U}_p$. Par hypothèse de récurrence, elle est définissable par un terme t, on voit facilement que σ est définissable par $\lambda x_1 \ldots \lambda x_n.x_i;(t)x_1 \ldots x_n$.
- Si la réponse de σ est q_i et si $A_i \neq \mathbb{U}$, on note $A_i = B_1 \to \cdots \to B_m \to \mathbb{U}$ et $B_j = C^j_1 \to \cdots \to C^j_{r_j} \to \mathbb{U}$ (avec éventuellement $r_j = 0$) et q'_j le coup initial de B_j. Pour chaque $1 \leq j \leq m$, on définit la stratégie τ_j sur $(A_1 \to \cdots \to A_n \to C^j_1 \to \cdots \to C^j_{r_j} \to \mathbb{U}) + \mathbb{U}_1 + \cdots + \mathbb{U}_p + \mathbb{U}_{p+1}$ par ses vues : si $qq_iq'_js$ est une vue de σ, on obtient une vue de τ_j à partir de qs en faisant pointer sur q les coups de s qui pointaient sur q'_j et en remplaçant les coups $\checkmark$ qui pointaient sur q'_j par $\checkmark^0_{p+1}$. Par hypothèse de récurrence, τ_j est définissable par un terme t_j tel que $\vdash t_j : A_1 \to \cdots \to A_n \to C^j_1 \to \cdots \to C^j_{r_j} \to \mathbb{U} \mid \alpha_1 : \mathbb{U}_1, \ldots, \alpha_p : \mathbb{U}_p, \alpha_{p+1} : \mathbb{U}_{p+1}$.
  On note u le terme :
  $$\begin{array}{rl} u = (x_i) & \lambda y^1_1 \ldots \lambda y^1_{r_1}.\mu\alpha_{p+1}[\alpha_0](t_1)x_1 \ldots x_n y^1_1 \ldots y^1_{r_1} \\ & \vdots \\ & \lambda y^m_1 \ldots \lambda y^m_{r_m}.\mu\alpha_{p+1}[\alpha_0](t_m)x_1 \ldots x_n y^m_1 \ldots y^m_{r_m} \end{array}$$
  tel que $x_1 : A_1, \ldots, x_n : A_n \vdash u : \mathbb{U} \mid \alpha_0 : \mathbb{U}_0, \alpha_1 : \mathbb{U}_1, \ldots, \alpha_p : \mathbb{U}_p$.
  On s'interesse maintenant à la réponse de σ à la vue $qq_i\checkmark_i$, si c'est $\checkmark$ dans A ou $\checkmark^0_k$ dans $\mathbb{U}_k$, σ est définissable par $\lambda x_1 \ldots \lambda x_n.\mu\alpha_0[\alpha_k]u$ avec $k = 0$ pour $\checkmark$ dans A. Si la réponse de σ à $qq_i\checkmark_i$ est le coup $q_{i'}$ dans $A_{i'}$, on définit $\tau = \{\varepsilon\} \cup \{qs \mid qq_i\checkmark_i s \in \sigma\} : A + \mathbb{U}_1 + \cdots + \mathbb{U}_p$ qui est définissable, par hypothèse de récurrence, par un terme t tel que $\vdash t : A \mid \alpha_1 : \mathbb{U}_1, \ldots, \alpha_p : \mathbb{U}_p$. On montre que σ est définissable par $\lambda x_1 \ldots \lambda x_n.\mu\alpha_0[\alpha_0](u;(t)x_1 \ldots x_n)$. □

Exercice 28 (Complétude avec $\mathbb{B}$)
Démontrer le théorème précédent pour un type construit à partir de $\mathbb{B}$, par rapport au $\lambda\mu$-calcul enrichi avec la primitive $\mathsf{if} \ldots \mathsf{then} \ldots \mathsf{else} \ldots$ et les deux constantes V et F.

2.5 Isomorphismes de types

L'objet de cette section est d'appliquer notre modèle de jeux du $\lambda\mu$-calcul à une question purement syntaxique : la caractérisation des isomorphismes de types du $\lambda\mu$-calcul. L'utilisation du modèle permet de simplifier considérablement l'étude de cette question.

Pour une présentation des isomorphismes de types dans le cadre du λ-calcul, on pourra se référer au livre de R. Di Cosmo [6].

Définition 26 (Types isomorphes)
Deux types A et B sont dits *isomorphes*, noté $A \simeq_{\lambda\mu} B$, dans le $\lambda\mu$-calcul s'il existe deux $\lambda\mu$-termes u et v tels que :
- $\vdash u : A \to B \mid$
- $\vdash v : B \to A \mid$
- $\lambda x.(u)(v)x \simeq_{\beta\eta\mu\rho\theta} \lambda x.x$
- $\lambda y.(v)(u)y \simeq_{\beta\eta\mu\rho\theta} \lambda y.y$

Résoudre le problème des isomorphismes de types pour le $\lambda\mu$-calcul consiste à trouver d'autres caractérisations de la relation $A \simeq_{\lambda\mu} B$. On va pour cela utiliser le modèle de jeux. Ceci nous donne une application de la sémantique des jeux pour résoudre un problème a priori complètement indépendant comme le montre en particulier l'énoncé du théorème 4 qui ne mentionne que la syntaxe.

Définition 27 (Partie zig-zag)
Une partie s sur l'arène $A \to B$ est *zig-zag* si :
- chaque coup de Joueur qui suit un coup d'Opposant dans A (resp. B) est dans B (resp. A) ;
- chaque coup de Joueur dans A qui suit un coup initial d'Opposant dans B est justifié par celui-ci.

Si, de plus, $s \upharpoonright_A$ et $s \upharpoonright_B$ ont les mêmes pointeurs, s est dite *zig-zag avec pointeurs*.

Soit s une partie zig-zag de longueur paire sur l'arène $A \to B$, on note $\overline{s}$ l'unique partie zig-zag sur $B \to A$ telle que $\overline{s} \upharpoonright_A = s \upharpoonright_A$ et $\overline{s} \upharpoonright_B = s \upharpoonright_B$.

Exemple 30 (Identité)
Soit A une arène, les parties de id_A sont des parties zig-zag avec pointeurs.

Lemme 20 (Composition des parties zig-zag)
Si $s \in \mathcal{P}_{A \to B}$ est une partie zig-zag de longueur paire, il existe une interaction $\tilde{s} \in \mathrm{int}(B, A, B)$ telle que $\tilde{s} \upharpoonright_{B \to A} = \overline{s}$ et $\tilde{s} \upharpoonright_{A \to B} = s$.

Démonstration : On raisonne par récurrence sur la longueur de s. Si $s = \varepsilon$, $\overline{s} = \varepsilon$ et $\tilde{s} = \varepsilon$. Si $s = tmn$, par hypothèse de récurrence, il existe une interaction $\tilde{t}$ telle que $\tilde{t} \upharpoonright_{B \to A} = \overline{t}$ et $\tilde{t} \upharpoonright_{A \to B} = t$. Si m est dans A (donc n dans B) (resp. m est dans B et n dans A), on a $\tilde{s} = \tilde{t}n_1mn_2$ où n_1 est une copie de n dans le B de gauche et n_2 est une copie de n dans le B de droite (resp. $\tilde{s} = \tilde{t}m_2nm_1$). □

Proposition 6 (Isomorphismes de jeux)
Soient A et B deux arènes, A et B sont isomorphes dans le modèle de jeux, noté $A \simeq_{\mathcal{G}} B$, si et seulement si elles sont identiques (isomorphes en tant qu'ordres partiels), noté $A \equiv B$.

Démonstration : Soient $\sigma : A \to B$ et $\tau : B \to A$ les deux stratégies qui réalisent l'isomorphisme, on montre par récurrence sur l'entier pair k que si $s \in \sigma$ est de longueur k, alors s est une partie zig-zag avec pointeurs et $\{t \mid t \in \tau \wedge |t| = k\} = \{\overline{s} \mid s \in \sigma \wedge |s| = k\}$:

- Si $k = 0$ alors $s = \varepsilon$ et le résultat est évident.
- Si $k = k' + 2$ et $s = tmn$ est une partie de σ de longueur k, par hypothèse de récurrence, t est zig-zag avec pointeurs et $\overline{t} \in \tau$. Supposons que m est dans B (le cas m dans A se traite de la même manière), si $n \in B$, par le lemme 20, $\tilde{t}mn \in \mathrm{int}(B, A, B)$ est telle que $\tilde{t}mn \upharpoonright_{B \to A} = \tilde{t} \upharpoonright_{B \to A} = \overline{s} \in \tau$ et $\tilde{t}mn \upharpoonright_{A \to B} = \tilde{t} \upharpoonright_{A \to B} mn = tmn \in \sigma$. Ainsi $\tilde{t}mn \upharpoonright_{B \to B} \in \tau; \sigma = id_B$ ce qui est impossible car toute partie de id_B est zig-zag (exemple 30), d'où $n \in A$.

  Soit s' la partie de id_B telle que $s' \upharpoonright_{B_1} = s' \upharpoonright_{B_2} = s \upharpoonright_B = t \upharpoonright_B m$. Par définition de la composition $\tau; \sigma$, il existe $u \in \mathrm{int}(B, A, B)$ telle que $u \upharpoonright_{B \to B} = s'$, $u \upharpoonright_{B \to A} \in \tau$ et $u \upharpoonright_{A \to B} \in \sigma$. En utilisant la preuve du lemme 3, on en déduit que $u = \tilde{t}mnm$ et donc n est justifié par m dans s si m est initial (sinon $u \upharpoonright_{B \to B} \notin id_B$), d'où s est zig-zag. De plus $\tilde{t}mnm \upharpoonright_{B \to A} = \tilde{s} \upharpoonright_{B \to A} = \overline{s} \in \tau$ donc $\{\overline{s} \mid s \in \sigma \wedge |s| = k\} \subset \{s \mid s \in \tau \wedge |s| = k\}$. La réciproque est obtenue de la même manière.

  Concernant les pointeurs, n est justifié dans $s \upharpoonright_A$ par un coup qui se trouve avant celui qui justifie m dans $s \upharpoonright_B$ (par innocence de σ), plus précisément ce doit être le coup situé juste avant, sinon m ne serait pas justifié par un coup de $\ulcorner \overline{t}n \urcorner$ dans $\overline{t}nm \in \tau$ ce qui violerait l'innocence de τ.

On montre alors que σ et τ induisent un morphisme d'ordre de A dans B et un morphisme d'ordre de B dans A qui sont inverses l'un de l'autre. □

On considère l'extension naturelle du $\lambda\mu$-calcul avec des types produits.

La théorie $=_{\mathcal{C}}$ est la théorie équationelle sur les types simples avec produits engendrée par les équations :

$$\begin{aligned}
A \times B &=_{\mathcal{C}} B \times A \\
A \times (B \times C) &=_{\mathcal{C}} (A \times B) \times C \\
A \times \top &=_{\mathcal{C}} A \\
(A \times B) \to C &=_{\mathcal{C}} A \to (B \to C) \\
\top \to A &=_{\mathcal{C}} A \\
A \to (B \times C) &=_{\mathcal{C}} (A \to B) \times (A \to C) \\
A \to \top &=_{\mathcal{C}} \top
\end{aligned}$$

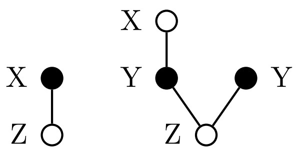
Lemme 21 (Formes canoniques)
En orientant les 5 dernières équations de la gauche vers la droite, tout type simple peut être mis, modulo $=_{\mathcal{C}}$, en une forme canonique : un produit (éventuellement vide) de types simples construits uniquement avec $\to$.

Pour résoudre la question des isomorphismes de types, on modifie l'interprétation des types simples avec variables en forêts, sans passer par une interprétation arbitraire de ces variables.

Définition 28 (Forêt étiquetée)
Une *forêt étiquetée* est la donnée d'une forêt et d'une fonction des nœuds de cette forêt dans l'ensemble des variables (X, Y, ...).

L'interprétation d'un type simple en forêt étiquetée est définie comme dans le cas des forêts excepté pour les variables : l'interprétation de X est la forêt à un nœud étiqueté par X.

Exemple 31
L'interprétation du type $(X \to Z) \times ((X \to Y) \to Y \to Z)$ est la forêt étiquetée suivante :



Théorème 4 (Isomorphismes classiques)
Soient A et B deux types simples construits avec les connecteurs $(\rightarrow, \times, \top)$ et des variables,

$$A \simeq_{\lambda\mu} B \quad \Longleftrightarrow \quad A =_{\mathcal{C}} B \quad \Longleftrightarrow \quad A \simeq_{\lambda} B$$

DÉMONSTRATION : Si $A \simeq_{\lambda\mu} B$, alors les stratégies associées aux $\lambda\mu$-termes qui réalisent cet isomorphisme forment un isomorphisme dans le modèle de jeux car leurs composées sont l'identité.

On en déduit que si $A \simeq_{\lambda\mu} B$, pour toute interprétation des variables de A et B par des forêts, les forêts associées sont identiques (proposition 6). On en déduit que les forêts étiquetées associées à A et B sont les mêmes (en interprétant la variable X_i par une chaîne de longueur $i(h+1)$ où h est la hauteur commune à A et B quand on interprète toutes les variables par $\perp$).

On montre, par récurrence sur la taille de la forêt étiquetée F associée à A et B, que $A =_{\mathcal{C}} B$. On suppose que A et B sont en forme canonique. Ce sont des produits de n types flèches où n est le nombre d'arbres de F. Chaque terme du produit A correspond alors à un terme du produit B et, par hypothèse de récurrence (si $n > 1$), ces termes sont égaux modulo $=_{\mathcal{C}}$ donc A et B aussi. Si $n = 1$, A et B sont tous les deux de la forme $C_1 \rightarrow \cdots \rightarrow C_k \rightarrow X$ où X est l'étiquette de la racine de F. On applique l'hypothèse de récurrence à la forêt étiquetée associée aux deux types $C_1 \times \cdots \times C_k$. Si $n = 0$, $A =_{\mathcal{C}} \top =_{\mathcal{C}} B$.

Si $A =_{\mathcal{C}} B$, il est facile de construire les λ-termes tels que $A \simeq_{\lambda} B$.

Si $A \simeq_{\lambda} B$ alors $A \simeq_{\lambda\mu} B$ car le λ-calcul est un sous-système du $\lambda\mu$-calcul. □

3 Questions, réponses et complétude pour le λ-calcul

Dans le but d'obtenir un résultat de complétude pour le λ-calcul, on va introduire une nouvelle contrainte sur les stratégies : le *bon parenthésage* et montrer que toute stratégie bien parenthésée est l'interprétation d'un λ-terme.

3.1 Arènes avec questions et réponses

Définition 29 (Feuille)
Soit A une arène, les coups de A qui n'ont pas de fils sont appelés les *feuilles* de A. Les feuilles qui ne sont pas des racines sont appelées des *feuilles strictes*.

En plus de l'information de polarité associée aux coups, on ajoute une distinction entre questions et réponses.

Définition 30 (Arène à réponses)
Une *arène à réponses* est une arène dans laquelle un sous-ensemble de ses feuilles strictes est distingué et constitue l'ensemble des réponses, notées `R`, les autres coups sont les questions, notées `Q`.

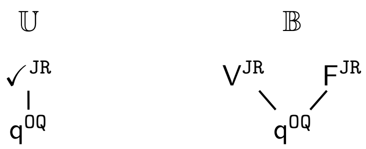
Les arènes que l'on a vues jusqu'ici apparaissent comme le cas particulier où tous les coups sont des questions.

On étend les constructions d'arènes pour définir leurs réponses.

Définition 31 (Constructions avec réponses)
Soient A et B deux arènes à réponses,

- l'arène $\top$ n'a pas de réponses ;
- l'arène $\perp$ n'a pas de réponses ;
- l'arène $A \times B$ a pour réponses l'union des réponses de A et des réponses de B ;
- l'arène $A \to B$ a pour réponses l'union des réponses de A et des réponses de B ;
- l'arène $A + B$ a pour réponses l'union des réponses de A et des réponses de B.

Exemple 32 (Arènes $\mathbb{U}$ et $\mathbb{B}$)
Dans les arènes $\mathbb{U}$ et $\mathbb{B}$, toutes les feuilles strictes sont des réponses :



Exercice 29
Montrer que, dans l'arène qui interprète un type simple construit avec $\mathbb{U}$, $\mathbb{B}$, $\to$ et $\times$, toutes les feuilles sont strictes et sont des réponses.

3.2 Parenthésage

L'adjonction de questions et de réponses dans les arènes nous permet d'introduire une nouvelle contrainte sur les stratégies.

Définition 32 (Question pendante)
Soit s une partie sur l'arène à réponses A, une question de s est *pendante* si aucune réponse qui suit n'est justifiée par cette question.

Définition 33 (Bon parenthésage)
Une partie s sur l'arène A est *bien parenthésée* si toute réponse de s pointe sur la dernière question pendante.

Une stratégie $\sigma : A$ est *bien parenthésée* si pour toute partie sm de σ telle que s est bien parenthésée, sm est bien parenthésée.

Remarque : Dans le cas d'une arène dont tous les coups sont des questions, toute stratégie est bien parenthésée.

Proposition 7 (Composition des stratégies bien parenthésées)
Soient $\sigma : A \to B$ et $\tau : B \to C$ deux stratégies bien parenthésées, la stratégie $\sigma;\tau : A \to C$ est bien parenthésée.

Démonstration : Soit $sm \in \sigma;\tau : A \to C$, on suppose que m est dans C (le cas où m est dans A se traite de la même manière). Si s est bien parenthésée, supposons que sm ne l'est pas. On note u une interaction telle que $u\restriction_{A\to C} = sm$, m_0 le coup de C qui justifie m et q la dernière question pendante située entre m_0 et m. Si q est dans C, on a une contradiction avec le fait que τ est bien parenthésée car $u\restriction_{B\to C}$ est dans τ et m n'est pas justifié par q.

Si q est dans A, on considère la suite t de coups de u dans B entre q et m. Si t contient une question, celle-ci doit justifier une réponse avant m (par bon parenthésage de τ) et cette réponse ne peut être que dans t. On obtient un appariement des questions et réponses de t mais comme t est de longueur impaire (par l'automate des états, voir section 1.3.2), il reste un coup non apparié qui ne peut alors être qu'une réponse r. Cela nous donne une contradiction car, par bon parenthésage de σ, r devrait être justifiée par q ce qui est impossible car q est dans A et r est dans B. □

Proposition 8 (Interprétation bien parenthésée du λ-calcul)
Soit t un λ-terme de type A, la stratégie sur A qui interprète t est une stratégie innocente finie totale et bien parenthésée.

DÉMONSTRATION : On peut soit montrer que les constructions de catégorie cartésienne fermée préservent le bon parenthésage, soit montrer que l'interprétation directe donnée section 1.4 fournit une stratégie bien parenthésée. □

3.3 Complétude

On va reprendre la preuve de complétude pour le $\lambda\mu$-calcul que l'on a faite à la section 2.4 et montrer que dans le cas d'une stratégie bien parenthésée, on obtient un λ-terme.

Lemme 22 (Vues bien parenthésées)
Soit σ une stratégie bien parenthésée et s une vue *appartenant à σ, s est bien parenthésée.*

DÉMONSTRATION : Par récurrence sur la longueur de s. Si $s = \varepsilon$, le résultat est immédiat. Si $s = tmn$, par hypothèse de récurrence, t est bien parenthésée et m pointe sur le dernier coup de t donc tm est bien parenthésée et, puisque σ est bien parenthésée, on en conclut que tmn est bien parenthésée. □

Théorème 5 (Complétude pour le λ-calcul)
Soit A un type simple contenant comme unique atome $\mathbb{U}$, si σ est une stratégie innocente finie totale bien parenthésée, il existe un λ-terme de type A dont σ est l'interprétation.

DÉMONSTRATION : On raisonne par récurrence sur $|\sigma|$. On note $A = A_1 \to \cdots \to A_n \to \mathbb{U}$ (avec éventuellement $n = 0$ pour $A = \mathbb{U}$), q le coup initial de A et $\checkmark$ la réponse justifiée par q dans A. Par déterminisme, σ a une unique réponse à q.

- Si c'est $\checkmark$, σ est l'interprétation du terme $\lambda x_1 \ldots \lambda x_n.\mathsf{c}$.
- Sinon, c'est le coup initial d'un des A_i que l'on note q_i. Si $A_i = \mathbb{U}$, la réponse d'Opposant ne peut être que $\checkmark_i$ dans A_i, soit τ la stratégie définie par $\tau = \{\varepsilon\} \cup \{qs \mid qq_i\checkmark_i s \in \sigma\} : A$. Par hypothèse de récurrence, elle est définissable par un terme t, on voit facilement que σ est définissable par $\lambda x_1 \ldots \lambda x_n.x_i;(t)x_1 \ldots x_n$.
- Si la réponse de σ est q_i et si $A_i \neq \mathbb{U}$, on note $A_i = B_1 \to \cdots \to B_m \to \mathbb{U}$ et $B_j = C_1^j \to \cdots \to C_{r_j}^j \to \mathbb{U}$ (avec éventuellement $r_j = 0$) et q'_j le coup initial de B_j. Pour chaque $1 \leq j \leq m$, on définit la stratégie τ_j sur $A_1 \to \cdots \to A_n \to C_1^j \to \cdots \to C_{r_j}^j \to \mathbb{U}$ par ses vues : si $qq_iq'_js$ est une vue de σ, on obtient une vue de τ_j à partir de qs en faisant pointer sur q les coups de s qui pointaient sur q'_j et en supprimant l'éventuel coup $\checkmark'_j$ qui pointait sur q'_j, ce qui préserve le bon parenthésage par le lemme 22. Par hypothèse de récurrence, τ_j est définissable par un terme t_j tel que $\vdash t_j : A_1 \to \cdots \to A_n \to C_1^j \to \cdots \to C_{r_j}^j \to \mathbb{U}$.

Si la réponse de σ à la partie $qq_i\checkmark_i$ est $\checkmark$ dans A, on montre que σ est définissable par : $\lambda x_1 \ldots \lambda x_n.(x_i)\ (t_1)x_1 \ldots x_n\ \ldots\ (t_m)x_1 \ldots x_n$. Sinon la réponse est $q_{i'}$ et on applique l'hypothèse de récurrence à la stratégie obtenue en remplaçant les parties $qq_i\checkmark_i q_{i'}s$ par $qq_{i'}s$ ce qui nous donne un λ-terme t de type A et on montre que σ est définissable par : $\lambda x_1 \ldots \lambda x_n.((x_i)\ (t_1)x_1 \ldots x_n\ \ldots\ (t_m)x_1 \ldots x_n);(t)x_1 \ldots x_n$. □

3.4 Factorisation

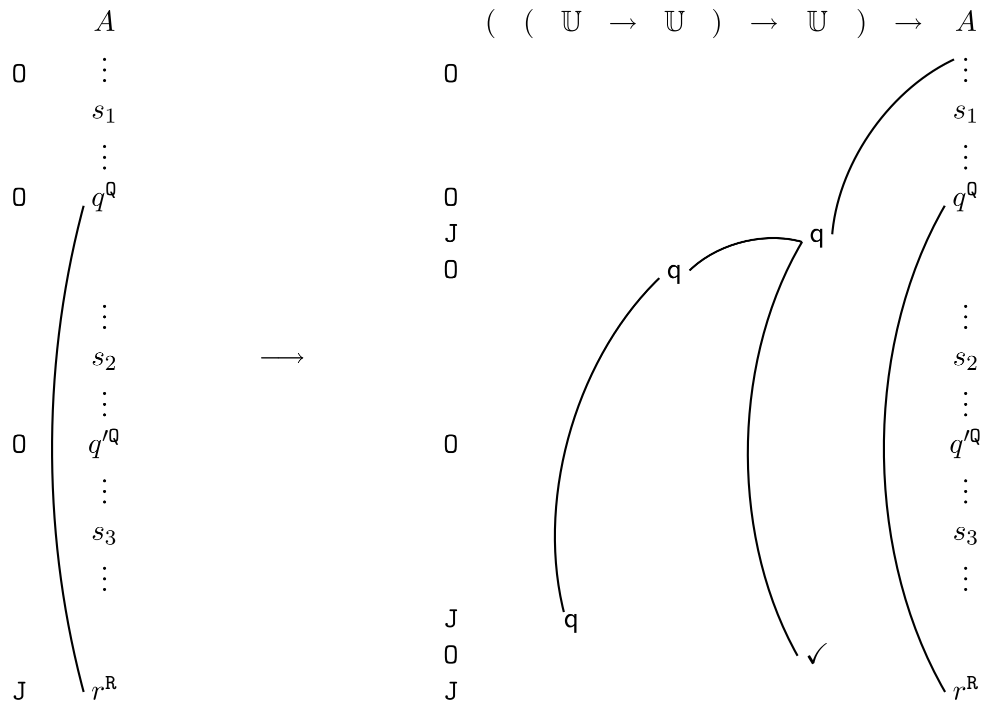
À partir de la preuve de complétude pour le λ-calcul, il est possible de prouver à nouveau la complétude pour le $\lambda\mu$-calcul par la méthode de *factorisation* [11].

Définition 34 (Catch)
Comme expliqué page 29 après l'exemple 24, les préfixes des parties suivantes :

permettent de définir une stratégie innocente finie totale, appelée catch. Cette stratégie n'est pas bien parenthésée.

Théorème 6 (Factorisation)
Soit $\sigma : A$ une stratégie innocente finie totale, il existe une stratégie τ innocente finie totale bien parenthésée sur l'arène $((\mathbb{U} \to \mathbb{U}) \to \mathbb{U}) \to A$ telle que $\sigma = \mathsf{catch};\tau$.

Démonstration : L'idée consiste à remplacer une partie de σ qui viole la condition de bon parenthésage par une partie sur $((\mathbb{U} \to \mathbb{U}) \to \mathbb{U}) \to A$ où Opposant viole le bon parenthésage et dont la partie gauche appartient à catch :



où $s_1 q s_2 q' s_3$ est bien parenthésée et q' est la dernière question pendante.

Soit s une vue maximale de σ, si s viole le bon parenthésage, on décompose s en $s_1 q s_2 q' s_3 r s_4$ où r est la première violation qui pointe sur q alors que la dernière question pendante est q' et on obtient deux vues : $s_1 q \mathsf{q} \mathsf{q} s_2 q' s_3 \mathsf{q}$ et $s_1 q \mathsf{q} \checkmark r s_4$. Si on obtient à nouveau une violation du bon parenthésage, on réapplique la même procédure jusqu'à obtenir une stratégie bien parenthésée τ. On vérifie facilement que $\mathsf{catch}; \tau = \sigma$. $\square$

Corollaire 6.1 (Complétude pour le $\lambda\mu$-calcul)
Soit A un type simple contenant comme unique atome $\mathbb{U}$, si σ est une stratégie innocente finie totale, il existe un $\lambda\mu$-terme de type A dont σ est l'interprétation.

Démonstration : On vérifie d'abord que $\mathsf{catch} : (\mathbb{U} \to \mathbb{U}) \to \mathbb{U}$ est définissable par le $\lambda\mu$-terme $\lambda f.\mu\alpha[\alpha](f)\mu\delta[\alpha]\mathsf{c}$. Par le théorème 6, il existe une stratégie τ innocente finie totale bien parenthésée telle que $\sigma = \mathsf{catch}; \tau$ et, par le théorème 5, τ est définissable par un λ-terme t de type $((\mathbb{U} \to \mathbb{U}) \to \mathbb{U}) \to A$. On en déduit que σ est l'interprétation de $(t)\lambda f.\mu\alpha[\alpha](f)\mu\delta[\alpha]\mathsf{c}$ de type A. $\square$

Références

[1] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Information and Computation*, 163(2) :409–470, December 2000.

[2] Samson Abramsky and Guy McCusker. Game semantics. In Helmut Schwichtenberg and Ulrich Berger, editors, *Logic and Computation : Proceedings of the 1997 Marktoberdorf Summer School*. Springer, 1998.

[3] Samson Abramsky and Guy McCusker. Full abstraction for idealized algol with passive expressions. *Theoretical Computer Science*, 227 :3–42, September 1999.

[4] Vincent Danos. An introduction to games semantics. Available at `http ://www.logique.jussieu.fr/www.danos/psgz/slides.ps.gz` , 2002.

[5] Vincent Danos and Russel Harmer. Probabilistic game semantics. *ACM Transactions on Computational Logic*, 3(3) :359–382, July 2002.

[6] Roberto Di Cosmo. *Isomorphisms of Types*. Progress in Theoretical Computer Science. Birkhäuser, 1995.

[7] Russel Harmer. *Games and Full Abstraction for Nondeterministic Languages*. Ph.D. thesis, Imperial College and University of London, 1999.

[8] Russel Harmer and Guy McCusker. A fully abstract game semantics for finite nondeterminism. In *Proceedings of the fourteenth annual symposium on Logic In Computer Science*, Trento, July 1999. IEEE, IEEE Computer Society Press.

[9] Martin Hyland and Luke Ong. On full abstraction for PCF. *Information and Computation*, 163(2) :285–408, December 2000.

[10] Jean-Louis Krivine. Un interpréteur du lambda-calcul. Available at `ftp ://ftp.logique.jussieu.fr/pub/distrib/krivine/interprt.pdf` , 1992.

[11] James Laird. Full abstraction for functional languages with control. In *Proceedings of the twelfth annual symposium on Logic In Computer Science*, pages 58–67, Warsaw, June 1997. IEEE, IEEE Computer Society Press.

[12] James Laird. *A semantic analysis of control*. Ph.D. thesis, University of Edinburgh, 1998.

[13] Olivier Laurent. Polarized games (extended abstract). In *Proceedings of the seventeenth annual symposium on Logic In Computer Science*, pages 265–274, Copenhagen, July 2002. IEEE, IEEE Computer Society Press.

[14] Olivier Laurent. Krivine's abstract machine and the $\lambda\mu$-calculus. Submitted for publication, September 2003.

[15] Olivier Laurent. Polarized proof-nets and $\lambda\mu$-calculus. *Theoretical Computer Science*, 290(1) :161–188, January 2003.

[16] Guy McCusker. *Games and Full Abstraction for a Functional Metalanguage with Recursive Types*. Ph.D. thesis, Imperial College and University of London, 1996. Published in Springer-Verlag's Distinguished Dissertations in Computer Science series, 1998.

[17] Michel Parigot. $\lambda\mu$-calculus : an algorithmic interpretation of classical natural deduction. In *Proceedings of International Conference on Logic Programming and Automated Reasoning*, volume 624 of *Lecture Notes in Computer Science*, pages 190–201. Springer, 1992.

[18] Dana Scott. A type-theoretical alternative to ISWIM, CUCH, OWHY. *Theoretical Computer Science*, 121 :411–440, December 1993.

[19] Peter Selinger. Control categories and duality : on the categorical semantics of the lambda-mu calculus. *Mathematical Structures in Computer Science*, 11(2) :207–260, April 2001.