Sémantique de Jeux et Décidabilité dans PCF

Pierre Clairambault

June 15, 2006

Contents

Abstract

On s'intéresse ici aux problèmes de décidabilité de l'équivalence contextuelle dans PCF (un λ-calcul simplement typé, muni d'un type de base et de constantes de ce type de base, ainsi que de connecteurs) et ses variantes. D'un côté on analyse les méthodes et résultats existants, de l'autre côté on applique, après les avoir introduits, les outils de la sémantique de jeux innocents [3] à l'extension de ces résultats de décidabilité.

1 Introduction

Ce présent rapport se situe entre deux problématiques.

L'une provient du monde syntaxique : V.padovani a démontré en 96 [9] que le *modèle minimal*[1] (noté Λ) a une équivalence contextuelle décidable. Etant donné deux termes de ce langage, on peut toujours savoir s'il existe un contexte qui les sépare. A l'opposé, R.Loader a montré [6] en 98 l'indécidabilité de la partie finie de PCF[2], notée PCF_2. Cette preuve était destinée à mettre fin à la question de l'existance d'un modèle concret pleinement adéquat pour PCF_2 : c'était impossible, l'égalité d'un tel modèle aurait nécessairement été indécidable puisqu'elle aurait du correspondre à l'équivalence contextuelle des termes.

En résumé, on était d'un côté décidable et de l'autre indécidable, sans qu'on puisse vraiment cerner la frontière. R.Harmer eut donc l'idée de considérer un langage intermédiaire, *sesqui*-PCF ($PCF_{1.5}$), possédant un connecteur $\wedge$ et une constante `ff` mais aucun moyen de l'éliminer : une exécution qui rencontre un `ff` se voit immédiatement interrompue, et renverra forcément `ff`. Résoudre le problème de l'équivalence contextuelle pour ce langage permettrait sans doute de mieux comprendre l'état des choses sur ses voisins proches. Une piste intéressante pour le résoudre est celle des *termes cellulaires* : des termes vérifiant certaines propriétés de structure, qui semblent jouer un rôle clef, explicitement ou non, dans toutes les preuves connues de décidabilité de Λ et de son extension immédiate par l'ajout de $\wedge$, PCF_1.

L'autre problématique provient de la sémantique de jeux, et intervient dans le sujet pour plusieurs raisons : la première, la plus naturelle, est que les jeux semblent formaliser bon nombre d'impressions intuitives qu'on a sur le calcul, notemment sur le rôle des connecteurs et des constantes. Là où l'ajout de ces connecteurs et constantes semble être anecdotique dans la syntaxe, il nécessite souvent dans les jeux de refondre complètement le modèle : une différence superficielle dans la syntaxe peut affecter profondément la sémantique, et ces modifications sembler mieux refléter la géométrie du calcul. On espère donc que ces lumières pourraient nous éclairer sur des points existants dans la syntaxe, mais qui passent inaperçus car inadaptés pour s'exprimer dans le monde syntaxique. La deuxième raison à l'intervention des jeux est encore une fois ces fameux termes cellulaires. En effet, au delà de leur intérêt pratique dans les questions de décidabilité, il se trouve qu'ils vérifient d'étonnantes propriétés en termes de jeux : ce sont les stratégies qui n'ont pas besoin d'arbitre pour intéragir, et leur processus d'interaction est une discussion directe.

On commencera donc, dans un premier temps, par présenter rapidement les différentes variantes de PCF dont les propriétés de décidabilité seront débattues au cours du rapport. Ensuite suivra une présentation de la sémantique de jeux innocente, qui répond au problème de la pleine adéquation pour PCF. C'est cette propriété de correspondance avec la syntaxe qui servira de base à la présentation, et on expliquera ensuite, à l'aide des intuitions acquises, les caractéristiques qui définissent habituellement la sémantique des jeux. Puis vient une partie qui mélange des travaux antérieurs (de V.Padovani) et des travaux récents, dédiée à la notion de cellularité. On la présentera dans sa formulation initiale, puis par ses multiples caractérisation et finalement on montrera comment on peut s'en servir pour décider l'équivalence contextuelle dans Λ. Finalement vient une partie entièrement nouvelle qui consiste en un développement d'un modèle général pour exprimer les connecteurs, c'est à dire les structures de contrôle locales. Ce modèle ne s'est malheureusement pas appliqué à $PCF_{1.5}$, mais est tout de même

[1] un λ-calcul simplement typé muni d'un unique type de base et de deux constantes de ce type de base, `tt` et $\perp$

[2] obtenue en rajoutant au précédent une constante `ff` et un `if _ then _ else`

parvenu à prolonger un peu la conaissance que nous avions de la décidabilité et à en expliquer les raisons.

D'autres travaux de ce stage n'ont pas pu être présentés ici, notemment un modèle de jeux présentant des *stratégies avec échec* et destiné à modéliser le langage $PCF_{1.5}$ où l'exécution peut être soudain interrompue par un échec. Les raisons en sont d'une part un manque de place, et d'autre part parceque les résultats de décidabilité n'en dépendaient pas directement, et c'est vers eux qu'à été axé ce rapport.

2 Préliminaires

2.1 PCF et ses variantes

Le langage PCF, d'abord introduit par [Plotkin], est un langage de programmation fonctionnel destiné, de par sa simplicité, à être étudié mathématiquement. C'est un λ-calcul simplement typé équipé d'entiers naturels, de récursion, et d'une sémantique opérationelle en appel par nom. L'équivalence observationelle (qui sera définie rigoureusement plus loin) est clairement indécidable dans PCF, cependant la question est loin d'être triviale pour la partie finie de PCF (notament sans les entiers naturels). On va commencer ici par donner une présentation succinte des variantes et restrictions finies de PCF qui seront évoquées tout au long du développement. Le lecteur pourra trouver en annexe une présentation exhaustive de la syntaxe et de la sémantique de PCF.

Notation: Une variable avec une flèche $\overrightarrow{x}$ signifie une suite finie de variables $x_1, \ldots, x_n$. De la même façon, on notera $f\overrightarrow{x}$ pour l'application successive de f aux variables $x_1 \ldots x_n$.

2.1.1 PCF fini, ou PCF_2

Definition 1. *PCF_2 est un λ-calcul simplement typé équipé d'un unique type de base noté o, et de trois constantes* t, f *et* $\perp$ *de ce type de base. On dispose également d'une construction* $\mathrm{if} \ldots \mathrm{then} \ldots \mathrm{else} \ldots$*, le tout muni de la même sémantique que dans PCF.*

Proposition 1. *La réduction dans PCF_2 est fortement normalisante et a la propriété de Church-Rosser. Ses formes normales sont de la forme:*

- *Les formes normales de type $A_1 \to \cdots \to A_n \to o$ sont les termes de la forme $\lambda x_1 \ldots x_n.r$ où $r : o$ est normal.*
- t, f *et* $\perp$ *sont normaux.*
- *Si f est une variable de type $A_1 \to \cdots \to A_n \to o$ ($n \geq 0$), si les s_i sont normaux de type A_i, et si $b, c : o$ sont normaux, alors $fs_1 \ldots s_n$ et* `if` $fs_1 \ldots s_n$ `then` b `else` c *sont normaux.*

En particulier, les formes normales closes de type o sont justes t, f *et* $\perp$.

2.1.2 Le modèle minimal, ou Λ

Definition 2. *Λ est un λ-calcul simplement typé équipé d'un unique type de base o, et de deux constantes de ce type de base :* t *et* $\perp$*. Sa sémantique opérationelle est la restriction de celle de PCF_2 qui lui correspond.*

Proposition 2. *La réduction dans Λ est fortement normalisante et a la propriété de Church-Rosser. Ses formes normales sont de la forme :*

- t *et* $\perp$ *sont normaux*
- *Si $\overrightarrow{u}$ sont des termes normaux de type $A_1, \ldots, A_n$, alors $\lambda\overrightarrow{y}.y_i\overrightarrow{u}$ est normal.*

En particulier, les formes normales closes de type o sont uniquement t *et* $\perp$.

2.1.3 PCF_1, ou PCF unaire

Definition 3. *PCF_1 est un λ-calcul simplement typé équipé d'un unique type de base o et de deux constantes de ce type de base* $\mathtt{tt}$ *et* $\perp$*, ainsi que d'un connecteur* $\wedge$*. La sémantique du* $\wedge$ *est la suivante:*

- $\mathtt{tt} \wedge b \to b$
- $\perp \wedge b \to \perp$

Proposition 3. *La réduction dans PCF_1 est fortement normalisante et a la propriété de Church-Rosser. Ses formes normales sont de la forme:*

- *Les formes normales de type $A_1 \to \cdots \to A_n \to o$ sont les termes de la forme $\lambda x_1 \ldots x_n.r$ où $r : o$ est normal.*
- t, f *et* $\perp$ *sont normaux.*
- *Si $\lambda \overrightarrow{y}.y_i\overrightarrow{u_i} : \overrightarrow{A} \to o$ et $\lambda \overrightarrow{y}.y_j\overrightarrow{u_j} : \overrightarrow{A} \to o$ sont formes normales (et sous forme η-longue), alors $\lambda \overrightarrow{y}.(y_i\overrightarrow{u_i}) \wedge (y_j\overrightarrow{u_j})$ est sous forme normale.*

En particulier, les formes normales closes de type o sont justes $\mathtt{tt}$ *et* $\perp$*.*

2.1.4 $PCF_{1.5}$, ou sesqui-PCF

Definition 4. *$PCF_{1.5}$ est un λ-calcul simplement typé équipé d'un unique type de base o et de trois constantes de ce type de base* $\mathtt{tt}, \mathtt{ff}$ *et* $\perp$*, ainsi que d'un connecteur* ;*. La sémantique du* ; *est la suivante:*

- $\mathtt{tt}; b \to b$
- $\perp; b \to \perp$
- $\mathtt{ff}; b \to \mathtt{ff}$

Proposition 4. *La réduction dans $PCF_{1.5}$ est fortement normalisante et a la propriété de Church-Rosser. Ses formes normales sont de la forme:*

- *Les formes normales de type $A_1 \to \cdots \to A_n \to o$ sont les termes de la forme $\lambda x_1 \ldots x_n.r$ où $r : o$ est normal.*
- t, f *et* $\perp$ *sont normaux.*
- *Si $\lambda \overrightarrow{y}.y_i\overrightarrow{u_i} : \overrightarrow{A} \to o$ et $\lambda \overrightarrow{y}.y_j\overrightarrow{u_j} : \overrightarrow{A} \to o$ sont formes normales (et sous forme η-longue), alors $\lambda \overrightarrow{y}.(y_i\overrightarrow{u_i}); (y_j\overrightarrow{u_j})$ est sous forme normale.*

En particulier, les formes normales closes de type o sont justes $\mathtt{tt}$ *et* $\perp$*.*

2.1.5 Plongements entre restrictions de PCF

Definition 5. ***(plongement)*** *Soient L_1, L_2 deux restrictions de PCF. On dit que L_1* ***se plonge*** *dans L_2 si et seulement si il existe une interprétation des constantes de L_1 dans celles de L_2 et des connecteurs de L_1 dans L_2 qui préserve la sémantique opérationelle, c'est à dire que la restriction de la sémantique opérationelle de L_2 à $F(L_1)$ doit correspondre exactement à la sémantique opérationelle de L_1.*

Examples.

- Λ se plonge dans PCF_1 : évident puisque ses constantes et connecteurs en sont un sous-ensemble.
- PCF_1 se plonge dans $PCF_{1.5}$: évident pour la même raison.
- $PCF_{1.5}$ se plonge dans PCF_2 : même raisons, on interprète $a; b$ par `if` a `then` b `else ff`.

2.2 Présentation informelle des Jeux Innocents

On va donner ici une présentation intuitive de la sémantique de Jeux, en ignorant les définitions formelles d'arène, de stratégie, d'innocence... En effet l'espace ici est trop restreint pour en rendre compte complètement, et ces définitions sont telles qu'en extraire l'intuition originale prend du temps et des efforts. On tentera ici de mettre en avant les intuitions, en illustrant par des exemples. La propriété cruciale que possèdent les jeux étant de fournir un modèle pleinement adéquat (càd, l'égalité y est équivalente à l'équivalence contextuelle des termes correspondant) de PCF, on va tenter de retrouver ici les notions des Jeux par l'analyse de ce langage et de ses restrictions.

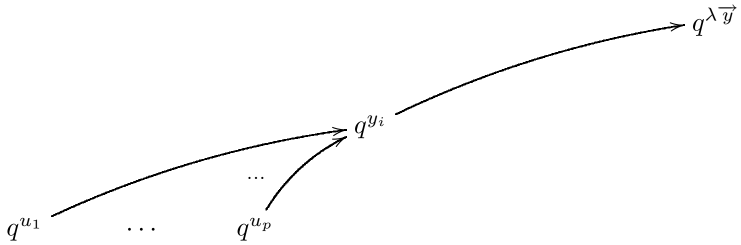
2.2.1 Structure de base d'une stratégie innocente

Afin de mettre la main sur les notions de base qui permettent de parler de termes de PCF avec des jeux, voyons ce que donnerait une interprétation en termes de jeu de la restriction la plus draconienne de PCF, Λ. L'idée est de considérer un terme sous forme normale (de tête), et de le caractériser non pas par *ce qu'il est* (ie. sa syntaxe), mais par *ce qu'il peut faire*, c'est à dire comment il peut réagir aux choix du joueur externe, qui joue pour les variables abstraites non encore substituées. L'intérêt de commencer par ce modèle est qu'il est d'une grande simplicité : il n'y a ici qu'un seul coup possible, la question. Il est inutile ici d'octroyer aux joueurs le droit de répondre étant donné qu'il n'y a aucun moyen de filtrer ces réponses : si une constante t ou $\perp$ apparait dans la partie, elle sera nécessairement le résultat final de la stratégie. Une stratégie qui en arrive à une telle constante se contentera donc de déclarer la partie terminée. Soit t un terme de Λ sous forme normale, t est nécessairement de la forme:

$$\lambda\overrightarrow{y}.y_i\overrightarrow{u} : A_1 \to \cdots \to A_n \to o$$

(avec A_i qui est lui-même de la forme $B_1 \to \cdots \to B_p \to o$). La stratégie qui correspond à notre terme est d'abord confrontée à une question initiale de l'environnement externe, incarné par l'*Opposant*, une question dont le sens pourrait être 'quelle est ta valeur?'. La réponse automatique du terme, incarné par le *Joueur*, est de poser une nouvelle question, cette fois à y_i. y_i est une variable opaque ne dépendant que de l'environnement; elle est donc incarnée par Opposant. Dès lors la partie peut se poursuivre de diverses manières : Opposant peut renvoyer t ou $\perp$ si y_i est une fonction constante, ou bien Opposant peut poser une nouvelle question à un des u_j si y_i est strict en sa j-ème variable, auquel cas ce sera une question initiale pour la stratégie correspondant au terme u_j. Ces choix non déterministes d'Opposant génèrent plusieurs branches dans la stratégie :

$$A_1 \longrightarrow \cdots \longrightarrow A_{i-1} \longrightarrow (B_1 \longrightarrow \cdots \longrightarrow B_p \longrightarrow o) \longrightarrow \cdots \longrightarrow A_n \longrightarrow o$$



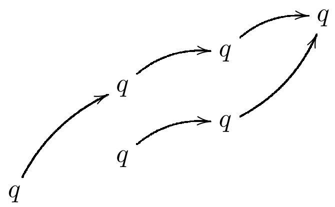
On obtient ainsi récursivement un ensemble de parties, ces parties ne pouvant différer que par des choix non déterministes de l'Opposant. La stratégie ainsi définie est une sorte de tableau des positions possibles où Joueur repère la situation actuelle et réagit en conséquence, par l'unique extension acceptable. Celà permet également de dégager la notion d'arène. En effet chaque coup de la partie correspond à un élément précis du terme, et donc appartient à un lieu précis dans le type.

Un élément reste à clarifier avant d'obtenir la sémantique habituelle :

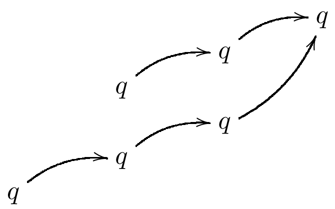
La Justification: Une partie est comme on l'a vu, un parcours précis dans le terme, ou plus précisément dans son arbre de Böhm. On pourrait donc penser que tout ce qui importe est l'ordre et la localisation des coups dans l'arène (*ie.* le type). Un autre élément est cependant important : *la justification.* Elle est incarnée par les flèches du diagramme ci-dessus, et correspond, suivant la dualité Opposant/Joueur, soit au lien entre une variable et son abstraction, soit au lien entre une fonction et son argument.

Exemple: Les stratégies associées aux deux *termes de Kierstead* (non égaux) ne diffèrent que par leur justifications:

$$K_x = \lambda f.f(\lambda x.f(\lambda y.x)) \qquad : \qquad ((o \longrightarrow o) \longrightarrow o) \longrightarrow o$$



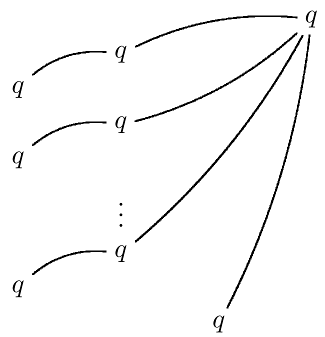
$$K_y = \lambda f.f(\lambda x.f(\lambda y.y)) \qquad : \qquad ((o \longrightarrow o) \longrightarrow o) \longrightarrow o$$

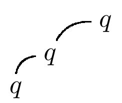


Ainsi, à partir d'un arbre de Böhm, on peut récupérer sa stratégie associée. Cette stratégie est complètement superposable à l'arbre de Böhm, et contient donc exactement les mêmes informations.

Exemple: On peut appliquer la transformation en stratégie décrite plus haut aux entiers de Church. On rappelle que l'entier de Church $\mathbf{n}$ est défini par $\mathbf{n} = \lambda fx.f^n x$, alors :

$$\mathbf{n} = \lambda fx.f^n x \qquad : \qquad (o \longrightarrow o) \longrightarrow o \longrightarrow o$$



avec n répétitions du motif  .

2.2.2 Interaction entre stratégies innocentes

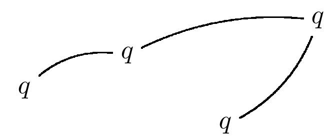
Les stratégies innocentes sont jusque là un nouveau moyen de représenter des termes de Λ, mais seul le côté statique a été dévoilé. En effet, l'intérêt principal de la sémantique de Jeux est qu'elle donne, au même titre que la Géométrie de l'Interaction[1], une sémantique de la dynamique du langage : la composition ne se fait pas par substitution, mais par une interaction explicite entre les deux stratégies.

Soient σ et τ deux stratégies jouant respectivement sur les arènes $A \Rightarrow B$ et $B \Rightarrow C$. L'interaction commence par une question initiale de l'Opposant externe posée à τ. τ va réagir en conséquence comme s'il était seul, mais dès lors qu'il jouera dans B c'est σ qui va prendre le relais et se substituer à l'Opposant externe. Tant que l'interaction a lieu dans B, τ jouera le rôle d'Opposant pour σ et réciproquement. Si jamais elle retourne dans A ou C, alors l'Opposant externe aura la main.

La stratégie composée est obtenue à partir de cette interaction en cachant toutes les communications internes qui ont eu lieu dans B. On obtient donc une stratégie $\sigma;\tau$ sur l'arène $A \Rightarrow C$ qui correspond à l'application des termes équivalents.

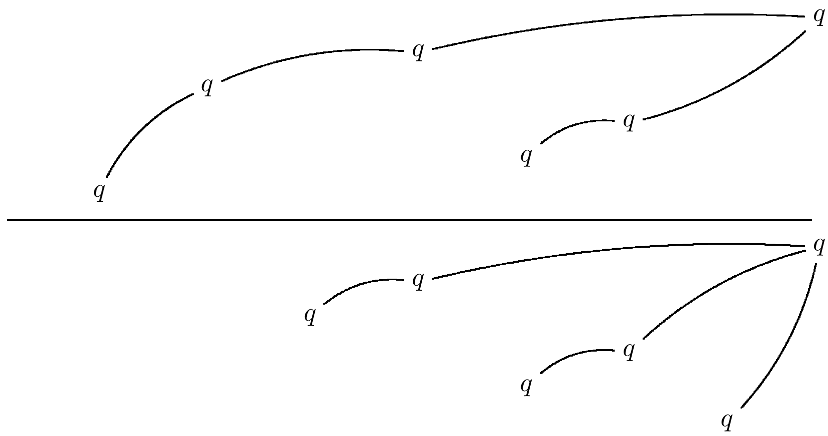
Exemple: Utilisons cette définition d'interaction pour calculer **S1**. Comme on l'a vu plus haut, la stratégie correspondant à **1** est:

$$\mathbf{1} = \lambda fx.fx \qquad : \qquad (o \longrightarrow o) \longrightarrow o \longrightarrow o$$



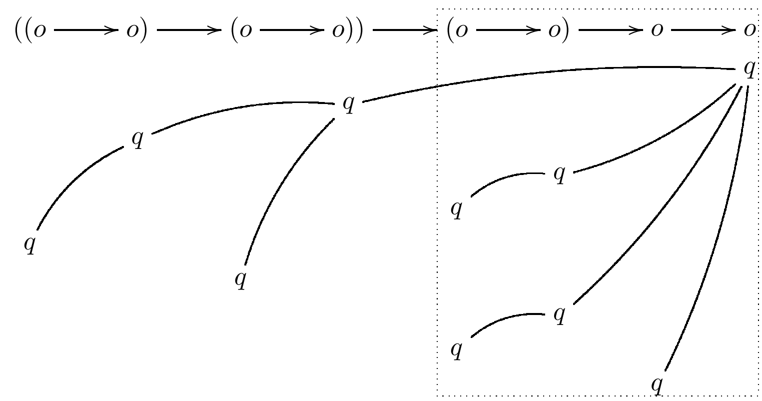
De même, la stratégie correspondant au successeur est :

$$\mathbf{S} = \lambda nfx.n(\lambda y.fy)(fx) \qquad : \qquad ((o \longrightarrow o) \longrightarrow (o \longrightarrow o)) \longrightarrow (o \longrightarrow o) \longrightarrow o \longrightarrow o$$



Ici, le trait horizontal permet une représentation des différents choix non déterministes d'Opposant. Ecrivons maintenant l'interaction entre les deux stratégies, en suivant les règles décrites plus haut : la stratégie **S** se substitue à l'Opposant externe "à gauche" pour **1**, et réciproquement:

S1 :



La partie encadrée de cette trace correspond à la partie de l'arène dans laquelle joue la stratégie **S1**. La stratégie issue de l'interaction est définie par la restriction de l'interaction à cette section de l'arène, et on obtient bien la stratégie correspondant à l'entier de Church **2**.

Théorème 1. ***(interaction vs. réduction)*** *L'interaction entre stratégies innocentes correspond exactement à la réduction normale[3] de tête des termes correspondants.*

2.2.3 Des Stratégies générales aux modèles de PCF

Au paragraphe précédent, on a défini les stratégies innocentes syntaxiquement par leur correspondance avec les termes, et on a vu que de telles stratégies pouvaient interagir entre elles, et que ces interactions correspondaient à la réduction (normale de tête) syntaxique. La démarche habituelle, plus formelle, consiste à définir les parties et les stratégies de manière générale et de poser des restrictions sur celles-ci pour récupérer les arbres de Böhm. Ces restrictions sont de deux ordres: l'**innocence** et le **bon parenthèsage**.

Innocence. L'innocence correspond au fait qu'une stratégie n'est pas consciente de l'ensemble de l'interaction. Dans un environnement purement fonctionnel, une fonction ne sait pas si elle a déjà été appellée auparavant, et ne peut tout simplement rien dire de l'execution courante sauf ce qui la concerne directement. L'innocence est simplement la formalisation de cette restriction de la vue qui bannit toute référence ou effet de bord du terme correspondant.

On impose à la stratégie innocente de ne dépendre que de sa *P-vue*, qui s'obtient en oubliant dans la partie toutes les sections se trouvant en dessous des pointeurs d'Opposant. Par exemple la P-vue d'une partie de la forme[4] :

$$\circ \; \bullet \; \cdots \; \circ \; \cdots \; \bullet \; \cdots \; \circ \; \bullet \; \circ \; \cdots \; \bullet \; \cdots \; \circ \; \cdots \; \bullet \; \circ$$

sera de la forme:

$$\circ \; \bullet \; \circ \; \cdots \; \bullet \; \circ \; \bullet \qquad\qquad \circ$$

On obtient donc une stratégie où les coups d'Opposant pointent toujours sur le coup précédent. Rappellons que les pointeurs d'Opposant correspondent au lien entre une fonction et son argument; la P-vue est alors la partie où l'Opposant n'appelle à chaque fois qu'un seul argument pour chaque fonction : il n'hésite jamais, et n'effectue aucun retour en arrière après avoir évalué un premier argument. Autrement dit, on ne sait rien de l'évaluation des autres arguments de la fonction, la seule chose qui compte est l'argument présent.

[3] il s'agit en fait d'une réduction *linéaire* de tête, qui simule la réduction normale de tête.

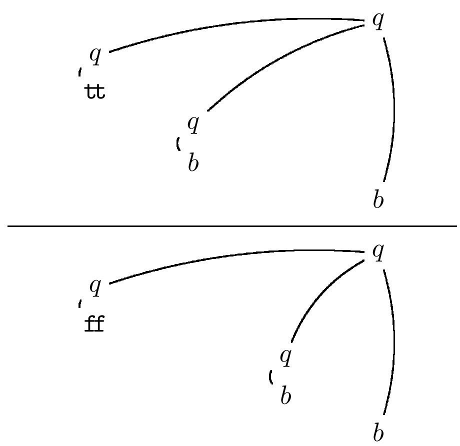
[4] Ici ○ représente les coups d'Opposant et ● les coups de Joueur. La partie se joue de gauche à droite, et on omet les justifications triviales.

Bon parenthèsage.

On a ébauché plus haut un modèle de jeux d'une version de PCF excessivement minimaliste, sans aucun connecteur. Les termes, à peu de chose près, ne sont constitués que de *combinateurs* qui ne peuvent pas filtrer sur les réponses de leurs calculs intermédiaires : leur comportement sera identique quel que soit le résultat des calculs intermédiaires. La seule façon pour une telle fontion de renvoyer un résultat est alors d'aller chercher, soit dans le terme soit dans l'environnement, une constante de type o et de la faire remonter. Une interaction entre de telles stratégies sera assimilable à un parcours simultané sur les deux arbres de Böhm, qui s'arrêtera dès que l'un des deux parcours tombera sur une constante (`tt` ou $\perp$).

Cependant, dès que le langage comprend des structures de contrôle telles qu'un `if _ then _ else` et plusieurs constantes de type de base, il devient nécessaire de répondre aux questions. Une réponse est tout simplement un coup dans la même partie de l'arêne que la question correspondante, et pointant vers cette même question. Dès lors il devient possible d'exprimer le `if_then_else`:

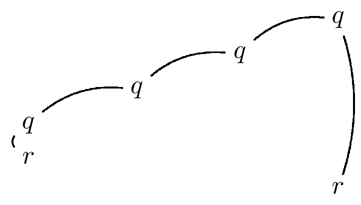
$\lambda b x_1 x_2.\texttt{if } b \texttt{ then } x_1 \texttt{ else } x_2 \quad : \quad o \longrightarrow o \longrightarrow o \longrightarrow o$



Avec cette nouvelle expressivité arrive une nouvelle contrainte : une partie sera valide seulement si elle est *bien parenthèsée*, c'est à dire qu'une réponse répond toujours à la dernière question posée. Cette contrainte exprime le caractère local du flot de contrôle : pour répondre à une question (un appel de fonction), il est nécessaire de répondre préalablement à toutes les autres questions posées depuis, c'est à dire de dépiler les appels de fonctions intermédiaires.

Remarquons qu'en oubliant cette contrainte de bon parenthèsage, il devient possible d'exprimer des structures de contrôle non locales, comme `call/cc`:

`call/cc` $\quad : \quad ((p \longrightarrow q) \longrightarrow p) \longrightarrow p$



2.3 Ordre Observationnel

On se place dans cette partie dans une restriction non spécifiée de PCF. On a donc un λ-calcul muni d'éventuelles constantes et structures de contrôle.

Definition 6. ***(ordre observationnel paramétrique)*** *Soient L une restriction de PCF. On peut toujours faire interagir ensemble des termes de n'importe quelle restriction de PCF puiqu'elles se plongent dans PCF complet.*
On définit l'ordre observationnel $\leq_L$ comme suit:

- *$x \leq_L y$ si et seulement si $x = \perp$ ou $y = x$, pour x, y deux constantes.*
- *$\leq_L$ est étendu aux termes clos de type o en comparant les formes normales*
- *$\leq_L$ est étendu aux termes clos de type fonctionnel $A \to B$ par $f \leq_L g$ si et seulement si, pour tous termes x, y dans L,*
$$x \leq_L y \Rightarrow fx \leq_L fy$$

$f \leq_L g$ doit se comprendre comme le fait que dans tout contexte de L, si f termine alors g aussi et les deux fonctions auront le même résultat. On en déduit la notion d'*équivalence observationnelle*:

Definition 7. ***(équivalence observationnelle paramétrique)***

$$a \equiv_L b \Leftrightarrow a \leq_L b \wedge b \leq_L a$$

Problème de décidabilité. On se pose alors la question de la décidabilité de l'égalité dans $L_1/\equiv_{L_2}$. Cette question provient de motivations diverses, mais les questions posées correspondent toujours au cas $L_1 = L_2$. Par exemple, le problème de la décidabilité de l'égalité dans $\Lambda/\equiv_\Lambda$ est équivalent au problème du filtrage atomique, c'est à dire à l'interpolation d'équations de la forme $f_i X_1^i \dots X_n^i = x_i$. La décidabilité en a été démontrée par V.Padovani. Ce résultat a été par la suite étendu à $PCF_1/\equiv_{PCF_1}$. De l'autre côté, le problème de l'égalité dans $PCF_2/\equiv_{PCF_2}$ a été inspiré par la recherche d'un modèle pleinement adéquat pour la partie finie de PCF : si l'égalité avait été décidable, il aurait été possible de donner concrètement ce modèle pleinement adéquat. Loader montra cependant que c'était impossible, et que l'égalité dans $PCF_2/\equiv_{PCF_2}$ était indécidable.

On aimerait comprendre en détail cette différence de décidabilité entre PCF_1 et PCF_2. Le processus de dichotomie nous amène directement à $PCF_{1.5}$, un langage hybride dont les caractéristiques refoulent toutes les attaques directes connues d'un côté et de l'autre. On tente donc un raisonnement plus fin : l'idée est de comprendre ces différences de décidabilité par une analyse des constantes et des structures de contrôle. Une première étape est de regarder au delà de la diagonale, en quotientant par d'autres équivalences que l'équivalence ambiante. Les travaux antérieurs peuvent être résumés ainsi :

	Λ	PCF_1	$PCF_{1.5}$	PCF_2	PCF
Λ	$+$				
PCF_1		$+$			
$PCF_{1.5}$					
PCF_2				$-$	
PCF					$-$

On tente d'abord de généraliser les preuves connues de décidabilité. Toutes ces preuves ont un point commun : elles s'appuyent toutes plus ou moins explicitement sur une certaines classe de termes, appellés par V.Padovani *termes cellulaires*. On commencera donc par étudier cette notion de cellularité qui semble être cruciale.

Equivalence contextuelle dans les Jeux. Avant de passer à la cellularité, évoquons le pendant de l'équivalence contextuelle des termes dans la sémantique de jeux.

Definition 8. ***(test)*** *Soit A une arène, un* **test** *sur A est une stratégie $\alpha : A \Rightarrow \Sigma$, où Σ est l'arène correspondant au type de base.*

Definition 9. ***(équivalence contextuelle)*** *Soient σ et τ deux stratégies dans une arène $A \Rightarrow B$, soit Δ un ensemble de* tests. *Alors $\sigma \equiv_\Delta \tau \Leftrightarrow \forall \alpha \in \Delta,\ \ulcorner \sigma \urcorner ; \alpha = \ulcorner \tau \urcorner ; \alpha$; où $\ulcorner \sigma \urcorner$ est l'équivalent* curryfié *de σ, qui joue dans l'arène $\mathbf{1} \Rightarrow (A \rightarrow B)$ plutôt que $A \Rightarrow B$, $\mathbf{1}$ étant l'arène vide où rien ne peut se passer.*

Cette définition est bien sur en adéquation avec l'équivalence contextuelle dans la syntaxe : si Δ est un ensemble de stratégies obtenues par interprétation d'un langage L, alors $\equiv_L$ sur les termes est en complète adéquation avec $\equiv_\Delta$ sur les stratégies.

3 Cellularité

3.1 Historique

Le concept de cellularité apparait pour la première fois dans [9] (Padovani, 96). Les termes cellulaires portent alors le nom de *termes transférants* et jouent un rôle clé dans le processus de décision de l'équivalence contextuelle de Λ. Quelques années plus tard apparait une preuve de décidabilité de PCF_1 [5] (Loader,98) où apparaissent les mêmes *termes transférants*. La décidabilité de PCF_1 est également montrée par Shmidt-Shauss [10] , sans recours apparent aux *termes transférants*. Sa méthode est reprise par Loader [4] pour donner la dernière version en date de la preuve de décidabilité de Λ, faisant moins de 4 pages. Dans une version révisée [8] de sa première preuve de décidabilité (non publiée), V.Padovani rebaptise ces termes *termes cellulaires* et montre qu'ils sont tout de même au coeur de la preuve de décidabilité de PCF_1 par Shmidt-Shauss. Plus tard R.Harmer remarque les propriétés inhabituelles des termes cellulaires dans la sémantique de jeux.

3.2 Définition et caractérisation

La définition a été initialement écrite dans Λ et on commencera ici par donner celle-là, cependant elle se généralise directement à toutes les restrictions de PCF en remplacant le cas de base de la définition par la forme normale des termes du langage considéré. Notons que d'autres caractérisations de la cellularité (données plus loin) semblent plus adroites, puisqu'elles sont invariantes suivant le langage considéré, cependant leur défaut est qu'elles passent sous silence le concept de *cellule*, qui est crucial pour comprendre la dynamique de ces termes.

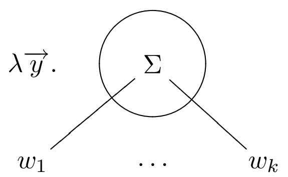
3.2.1 Termes cellulaires

Definition 10. ***(cellules)*** *Une* **cellule** *est un contexte de type de base avec des trous de type de base, de la forme $(yv_1 \dots v_p)[]_1 \dots []_K$, où $v_1, \dots, v_p$ sont des termes clos.*

Definition 11. ***(termes cellulaires)*** *On dit qu'un terme t est* **cellulaire** *si et seulement si il est de l'une des formes suivantes :*

- *$\lambda y_1 \dots y_n.a$ où a est une constante*
- *$\lambda y_1 \dots y_n.\Sigma[w_1]_1 \dots [w_k]_K$, où Σ est une cellule et où chacun des $\lambda y_1 \dots y_n.w_j$ est cellulaire.*

L'arbre de Böhm d'un terme cellulaire ressemble donc à :



Avec les w_j inductivement cellulaires.

3.2.2 Caractérisation dans les Jeux

La définition syntaxique des termes cellulaires peut sembler opaque, cependant ils admettent une caractérisation intuitive dans la sémantique des Jeux:

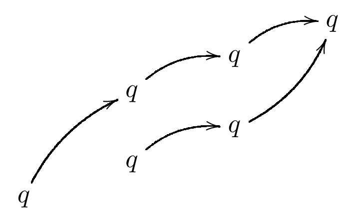
Definition 12. ***(OP-vue)****Soit s une partie. On obtient l'****OP-vue*** *de s en suivant les pointeurs des deux joueurs jusqu'au coup initial en en gardant tous les coups rencontrés.*

Proposition 5. ***(stratégies innocentes cellulaires)*** *Soit t un terme de Λ, on l'interprète dans la sémantique des jeux par une stratégie σ innocente. Alors les trois propositions suivantes sont équivalentes:*

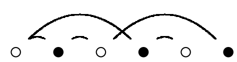
- *t est cellulaire*
- *Il n'y a pas de croisement entre des pointeurs de Joueur dans σ.*
- *Tous les pointeurs de Joueur dans σ pointent dans leur OP-vue.*

Exemple: Rappellons la stratégie liée au terme K_x de Kierstead.

$$K_x = \lambda f.f(\lambda x.f(\lambda y.x)) \qquad : \qquad ((o \longrightarrow o) \longrightarrow o) \longrightarrow o$$



Ce terme n'est pas cellulaire. En effet mettons à plat la stratégie, en oubliant la nature des coups et en gardant juste jeur polarité ainsi que les justifications. On obtient :



Et il apparait bien un croisement entre deux pointeurs de Joueur. Toute tentative de mettre le terme K_x sous forme cellulaire se heurtera au même problème : les sous-termes insérés dans les trous de la première cellule seront bien de type de base, mais ne pourront être clos. Si on avait remplacé le x par un y pour obtenir K_y, on aurait cette fois-ci un terme cellulaire, c'est à dire une stratégie où tous les coups du Joueur pointent dans l'OP-vue.

Remarque: On peut simuler l'interaction innocente générale par l'action d'une machine abstraite qui maintient l'information globale sur l'interaction et interroge les deux stratégies en leur donnant la vue qu'elles ont sur la partie en cours (voir les travaux récents de R.Harmer). Une des propriétés des stratégies cellulaires est qu'il n'est pas nécessaire d'entretenir cette structure d'information quand elles interagissent. En conséquence, l'interaction entre des stratégies cellulaires peut être résolue avec une mémoire constante.

3.2.3 Preuves cellulaires

Dans cette sous-partie on se placera dans Λ par souci de simplicité, même si les résultats développés ici sont exprimables facilement en rajoutant des opérateurs de contrôle. Rappelons en premier lieu les règles de formation des arbres de Böhm intuitionnistes, autrement dit des termes de Λ sous forme $\beta\eta$-normale.

Definition 13. ***(IBT)***

$$\frac{}{\Gamma \vdash \bot : o} \qquad\qquad \frac{}{\Gamma \vdash \mathtt{tt} : o}$$

$$\frac{\Gamma, x : \overrightarrow{A} \to o \vdash M_1 : A_1 \quad \ldots \quad \Gamma, x : \overrightarrow{A} \to o \vdash M_n : A_n}{\Gamma, x : \overrightarrow{A} \to o \vdash x\ \overrightarrow{M} : o}$$

$$\frac{\Gamma, x_1 : A_1, \ldots, x_n : A_n \vdash M : o}{\Gamma \vdash \lambda \overrightarrow{x}.M : \overrightarrow{A} \to o}$$

Rappelons maintenant que :

- Les stratégies innocentes sont superposables aux arbres de Böhm
- La condition de cellularité dans les jeux est qu'il n'y aie pas de croisement entre deux pointeurs de Joueur
- Les pointeurs de Joueur correspondent au lien entre une variable et son abstraction

Ces trois éléments laissent entrevoir une caractérisation élégante des arbres de Böhm cellulaires : à peu de chose près, la dernière variable abstraite doit être la première utilisée. Ceci nous suggère des preuves où le contexte est assimilable à une pile. Pour accéder à un élément du contexte et utiliser un axiome, il est nécessaire de dépiler toutes les hypothèses abstraites auparavant.
Bien entendu, on refuse en général la commutativité à l'intérieur du contexte mais il nous en faut tout de même une forme restreinte puisque des variables abstraites au même point de l'arbre de Böhm sont utilisables dans n'importe quel ordre.

On considère donc des contextes munis de deux séparateurs : un séparateur ”,” commutatif, séparant des variables abstraites au même point, et un séparateur ”;” non commutatif.

Definition 14. ***(CellBT)***

$$\frac{}{\Gamma \vdash \bot : o} \qquad\qquad \frac{}{\Gamma \vdash \mathtt{tt} : o}$$

$$\frac{\Gamma_1, x : \overrightarrow{A} \to o \vdash M_1 : A_1 \quad \ldots \quad \Gamma_1, x : \overrightarrow{A} \to o \vdash M_n : A_n}{\Gamma_1, x : \overrightarrow{A} \to o ; \Gamma_2 \vdash x\ \overrightarrow{M} : o}$$

$$\frac{\Gamma; x_1 : A_1, \ldots, x_n : A_n \vdash M : o}{\Gamma \vdash \lambda \overrightarrow{x}.M : \overrightarrow{A} \to o}$$

Proposition 6. *Les preuves dérivées en* ***CellBT*** *correspondent exactement aux termes cellulaires.*

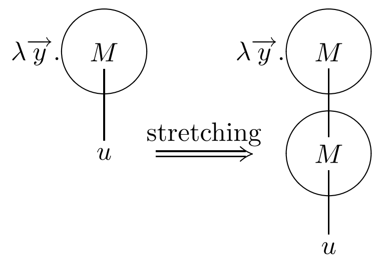
Proof. Tous les termes définis par ce système sont cellulaires. En effet par définition on interdit ici les pointeurs croisés, ce qui est une caractérisation de la cellularité. De la même manière tous les termes cellulaires sont définissables avec ce système, il suffit d'agencer correctement les contextes afin de perdre le minimum possible d'information lors de l'utilisation des variables, en fait de déplacer chaque variable le plus à droite possible dans le contexte lors de son utilisation. □

3.3 Cellularisation

Le contenu des deux parties qui suivent provient de discussions avec V.Padovani et n'a jamais été publié dans cette formulation. Dans cette partie, on se placera uniquement dans Λ pour y décrire le processus de *cellularisation*, qui donne un équivalent cellulaire à chaque terme et assurera ainsi l'universalité des termes cellulaires. Cette propriété est une des hypothèses clé nécessitées par l'algorithme de décision.
L'algorithme de cellularisation est défini par induction à partir de deux lemmes intermédiaires :

Lemme 1. ***(stretching)*** *Soit $t = \lambda y_1 \ldots y_n.M[u]$ un terme clos, où M est un contexte avec un trou de type de base. Alors t est contextuellement équivalent à $\lambda y_1 \ldots y_n.M[M[u]]$.*

Illustration. L'effet du stretching sur un arbre de Böhm peut être résumé par le schéma suivant:



L'intuition derrière ce lemme est simple : le contexte M n'a qu'un unique trou, de type de base. Par conséquent lors de son évaluation, seuls deux cas sont possibles:

- Il n'évalue pas le contenu de son trou, auquel cas les deux termes sont clairement équivalents
- Il évalue le contenu, mais dans ce cas, comme le trou est de type de base, il renverra nécessairement le résultat de l'évaluation de son trou. Les deux termes se comporteront donc de la même façon.

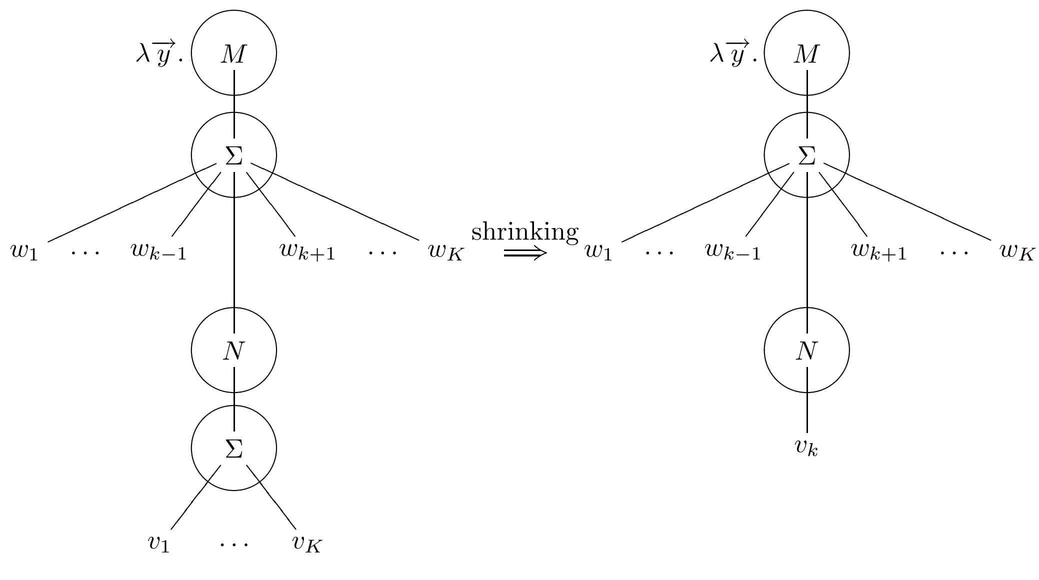
Lemme 2. ***(shrinking)*** *Soit Σ une cellule dont la variable de tête est l'un des $\{y_1, \ldots, y_n\}$. Alors le terme*

$$\lambda y_1 \ldots y_n.M[\Sigma[w_1] \ldots [w_{k-1}][N[\Sigma[v_1] \ldots [v_K]][w_{k+1}] \ldots [w_K]]$$

est contextuellement équivalent au terme

$$\lambda y_1 \ldots y_n.M[\Sigma[w_1] \ldots [w_{k-1}][N[v_k]][w_{k+1}] \ldots [w_K]]$$

Illustration. L'effet du shrinking sur un arbre de Böhm peut être résumé par le schéma suivant:



Ici, on prend en compte l'innocence de l'environnement : dans le premier terme, le Σ du bas s'évalue exactement dans le même contexte que celui du haut puisqu'il dépend des mêmes abstractions. Or on sait que le Σ du haut a fait appel à son k-ième trou. On est dans le même contexte, ce qui pousse l'Opposant externe, étant donné son innocence, à faire exactement les mêmes choix que précédemment. Par conséquent Σ fera encore appel à son k-ième argument, d'où l'équivalence proposée. Notons que contre un adversaire non innocent, c'est à dire un terme avec des références, ceci est clairement faux puisque Σ peut "se souvenir" qu'il a été appellé auparavant et se comporter différemment. En fait, le *shrinking* n'est vrai que dans Λ, et devient faux dans quasiment toutes ses extensions imaginables.

Definition 15. ***(termes semi-cellulaires)*** *On dit qu'un terme t est* ***semi-cellulaire*** *si et seulement si il est cellulaire, où qu'il est de la forme* $\lambda\overrightarrow{y}.y_i\overrightarrow{u}$*, où chaque* $\lambda\overrightarrow{y}.u_i$ *est cellulaire.*

Les termes semi-cellulaires sont en quelque sorte les candidats à la cellularité. Ils peuvent encore la rompre, mais seulement par des occurences d'un y_j dans les u_i dont les pointeurs peuvent entrer en collision avec des pointeurs de variables abstraites en tête de u_i.

Lemme 3. *Tout terme semi-cellulaire est contextuellement équivalent à un terme cellulaire*

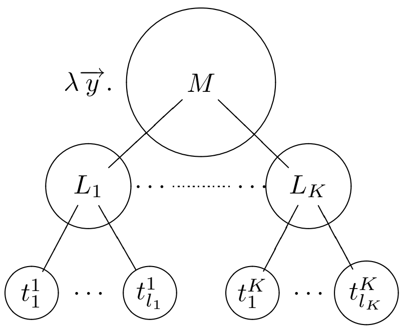
Proof. On raisonne par induction sur la longueur du terme t. Supposons que $t = \lambda\overrightarrow{y}.y_i\overrightarrow{u}$ est sous forme normale. Soit M le contexte minimal tel que:

- $M \neq []$
- $u = M[t^1]\dots[t^K]$
- chaque t^k est un terme de type de base, de la forme $(y\dots)$, avec $y \in \{y_1, \dots, y_n\}$.

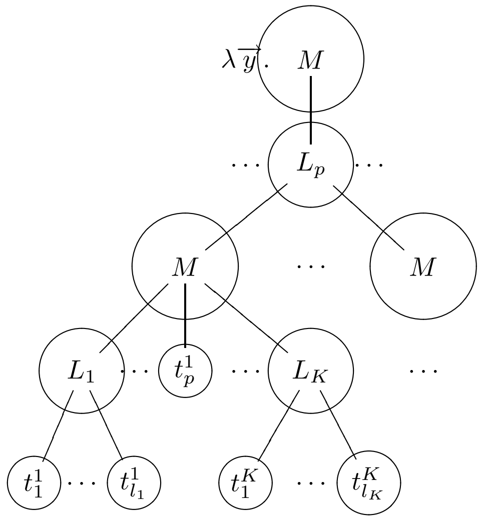
Pour chaque t^k, soit L_k le plus petit contexte tel que:

- $L_k \neq []$
- $t^k = L_k[t'^k_1]\dots[t'^k_{l_k}]$
- chaque t'^k_i est un terme de type de base de la forme $(z\dots)$, avec z abstrait dans un des u_i.

Pour résumer, on a mis le terme semi-cellulaire sous la forme :



On commence par effectuer un gigantesque *stretching* et remplacer tous les t^i_j par l'ensemble du terme. Puis on effectue un shrinking dans chaque copie, qu'on peut effectuer puisque dans chaque branche le contexte M est rencontré deux fois. On obtient :



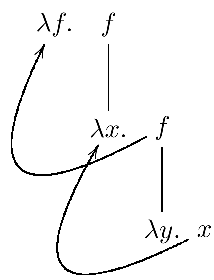
Tous les sous-termes commencant par une deuxième occurence de M sont alors semi-cellulaires mais sont de taille strictement inférieure à l'original, par conséquent l'hypothèse de récurrence nous en donne des équivalents cellulaires. Il nous suffit alors de recoler pour obtenir un équivalent cellulaire au terme tout entier. □

Théorème 2. ***(cellularisation)*** *Tout terme clos t est équivalent à un terme cellulaire.*

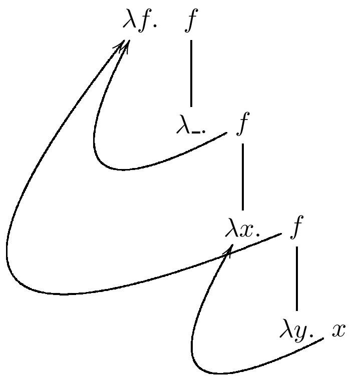
Proof. On raisonne par induction sur la longueur de t. Si $t = \lambda\overrightarrow{y}.(y_i\,\overrightarrow{u})$ alors par hypothèse d'induction, il existe des termes $u'_1, \ldots, u'_n$ tels que pour tout i, $\lambda\overrightarrow{y}.u'_i$ est un terme cellulaire équivalent à $\lambda\overrightarrow{y}.u_i$.

Le terme $t' = \lambda\overrightarrow{y}.(y_i\,\overrightarrow{u'})$ est alors semi-cellulaire, et équivalent à t. La conclusion découle alors du lemme. □

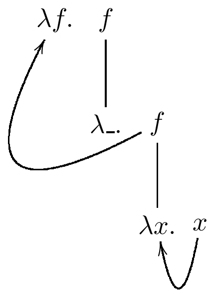
Example. Cet algorithme de cellularisation fait exploser la taille du terme, cependant il reste très raisonnable sur des termes de petite taille. Considérons le terme $K_x = \lambda f.f(\lambda x.f(\lambda y.x))$ non cellulaire. Son arbre de Böhm (où on rajoute les pointeurs de justification pour mettre en évidence la non-cellularité) est:



Ce terme est semi-cellulaire. On commence donc par expandre le terme par un stretching:



Puis on exploite la redondance générée en effectuant un shrinking.



Le terme obtenu $K'_x = \lambda f.f(\lambda d.f(\lambda x.x))$ est cellulaire, et contextuellement équivalent à l'original.

3.4 Résultats de décidabilité

L'universalité des termes cellulaires permet dans Λ de déduire la décidabilité de l'équivalence contextuelle.

Lemme 4. ***(Algorithme de listing)*** *Il existe une fonction R telle que pour tout type A, et pour tout ensemble C de constantes de type de base, $R(A, C)$ est une liste finie de terme comportant un représentant de chaque classe d'équivalence, pour la restriction de l'équivalence contextuelle aux termes clos dont les constantes appartiennent à C.*

Proof. Soit $A = A_1 \to \cdots \to A_n \to o$ où $A_i = B_1^i \to \cdots \to B_{p_i}^i \to o$. Soit $y_1 : A_1, \ldots, y_n : A_n$. Pour tout $i \in \{1, \ldots, n\}$ soit K_i un entier plus grand que le nombre de classes d'équivalence de type A_i. Pour tout $j \in \{1, \ldots, p_i\}$, on obtient récursivement $W_i^j = R(B_j^i, C \bigcup \{d_1^i \ldots d_{K_i}^i\})$, où les d_k^i sont des constantes fraiches de type de base. On pose $V_i = \{y_i \overrightarrow{w} | \overrightarrow{w} \in \Pi_{j=1}^{p_i}\}$. Alors $R(A, C)$ est défini comme le plus petit ensemble R vérifiant:

- $\lambda \overrightarrow{y}.a \in R$ ($a \in C$).
- Si $\lambda \overrightarrow{y}.w_1, \ldots, \lambda \overrightarrow{y}.w_{K_i} \in R$, et si $v \in V_i$ n'est pas utilisé dans la construction de ces termes, alors $\lambda \overrightarrow{y}.v[w_1/d_1^i] \ldots [w_{K_i}/d_{K_i}^i] \in R$.

On obtient un ensemble fini et dont tous les éléments sont cellulaires. Il reste à prouver qu'il atteint toutes les classes d'équivalences, ce qui ne sera pas fait ici en détail, la preuve étant technique et n'éclairant pas l'intuition. Elle suit le schéma suivant : on prend un terme quelconque de type A. Par cellularisation on peut supposer que t est cellulaire, ce qui impose suffisament sur se forme pour pouvoir l'associer à un terme de R équivalent, après d'éventuelles transformations qui préservent l'équivalence. $\square$

Théorème 3. ***(décidabilité de l'équivalence)*** *L'équivalence contextuelle est décidable dans* Λ.

Proof. L'algorithme de décision est simple une fois qu'on a l'algorithme de listing : soient $t_1, t_2 : A_1 \to \cdots \to A_n \to o$. Pour chaque i, on récupère la liste finie $R(A_i, \emptyset)$ puis on évalue t_1 et t_2 dans tous les contextes possibles (il y en a un nombre fini). $\square$

4 Jeux avec Connecteurs

Jusqu'à présent, on a considéré que la sémantique de jeux de Λ, qui ne comporte que des questions et pas de réponses. La nécessité des réponses apparait, comme on l'a vu, dès qu'on donne à un terme la capacité de se comporter différemment (dynamiquement) selon l'évaluation d'un sous-terme du type de base. En particulier, si on rajoute le connecteur $\wedge$ à Λ pour atteindre PCF_1 on aura besoin de réponses en terme de jeux. Le développement de cette partie vient de la remarque que ces réponses ne sont peut-être pas nécessaires pour PCF_1. En effet lorsqu'un terme appelle $M \wedge N$, on peut bien sur dire qu'il attend le retour de M pour décider si oui ou non il va évaluer N, dans ce cas des réponses sont nécessaires. L'autre point de vue est qu'il *sépare le flot de contrôle*, et évalue les deux à la fois, ce qui correspond à une sorte de concurrence sans synchronisation : la forme normale du terme dépendra alors de la réduction indépendante de chacune des deux branches.

En termes de jeux, nos parties ne seront plus représentées par des chaînes de coups avec des pointeurs, mais par des arbres de coups munis de pointeurs. Les stratégies seront donc des *bi-arbres*, c'est à dire qu'elles ont deux structures d'arbres associées : certains branchements sont toujours dûs au non déterminisme de l'Opposant externe, et certains autres sont dûs à la multiplication des flots de contrôle, c'est à dire les connecteurs.

Cette partie consistant en un développement entièrement nouveau du modèle des jeux, il serait difficile de le présenter sans le vocabulaire complet des jeux. Les notions seront tout de même définies ici, et on espère que l'intuition acquise précédemment aidera à leur compréhension.

4.1 Définitions

4.1.1 Arbres pointés

Definition 16. ***(arbres pointés)*** *Soit* Σ *un ensemble dénombrable, et* Δ *un ensemble fini de* connecteurs*. La structure de base est celle d'un arbre étiqueté par* Σ*, à l'exception des embranchements qui sont étiquetés par* Δ*. Pour une occurence* $s \in \Sigma$ *dans l'arbre, on note* $\overline{s}$ *la chaîne correspondant au chemin de la racine à* s*. Un* ***arbre pointé*** *est un arbre muni de pointeurs entre les noeuds (et feuilles) qui ne sont pas des embranchements, tels que si* s_i *pointe vers* s_j*, alors* $\overline{s_j}$ *est un préfixe propre de* $\overline{s_i}$*. De plus, on a* au plus un *pointeur sortant pour chaque noeud (ou feuille).*

Remarque: On peut voir un arbre pointé comme un ensemble de chaînes pointées, dont on a fusionné certain préfixes communs.

Definition 17. ***(arbre préfixe)*** *On appelle* ***arbre préfixe*** *d'un arbre pointé* s *tout arbre* t *inclus (en préservant la consécutivité et les pointeurs) dans* s *et contenant la racine.*

Definition 18. ***(restriction)*** *Soit* $\Sigma' \subseteq \Sigma$*, alors on obtient la* ***restriction de*** s ***à*** Σ'*, notée* $s_{\upharpoonright \Sigma'}$*, de la façon suivante : on retire de* s *les occurrences d'éléments de* $\Sigma - \Sigma'$*. De plus, on manipule les pointeurs comme suit: si* $s_i \in \Sigma'$ *pointe vers* $s_j \in \Sigma'$*, on garde le pointeur dans* $s_{\upharpoonright \Sigma'}$*. Sinon, on suit l'éventuel pointeur de* s_j *à un certain* s_k*, et on poursuit le même procédé récursivement. En d'autres termes, on suit les pointeurs de* $s_i \in \Sigma'$ *dans* Σ *jusqu'à ce qu'on émerge hors de* Σ *dans un* $s_z \in \Sigma$*, auquel cas on a que* s_i *pointe vers* s_z *dans* $s_{\upharpoonright \Sigma'}$*. Dans le cas où on émerge pas de* Σ*, alors* s_i *n'aura plus de pointeur.*

Remarque: Encore une fois cette définition revient à celle sur les chaînes pointées, puisque prendre la restriction à un certain Σ' d'un arbre, c'est exactement prendre séparément la restriction de toutes les chaînes pointées que consituent ses chemins.

4.1.2 Arènes

Definition 19. ***(arène)*** *Une* ***arène*** *A est un quadruplet $< M_A, \lambda_A, I_A, \vdash_A >$ où*

- *M_A est un ensemble dénombrable de* coups
- *$\lambda_A : M_A \to \{O, P\} \times \{Q, A\}$ étiquette chaque $m \in M_A$ comme un coup de Joueur ou d'Opposant, et comme une Question ou une Reponse. On écrit λ_A^{OP} (resp. λ_A^{QA}) sa composition avec la première (resp. seconde) projection, et λ_A^{PO} sa composition avec la première projection plus l'échange de O et P.*
- *I_A est un sous-ensemble de $\lambda_A^{-1}(OQ)$ nommé ensemble des* coups initiaux *de A.*
- *$\vdash_A$ est une relation binaire de* validation *sur M_A vérifiant:*
  - *Si $m \vdash_A n$ alors $\lambda_A(m) \neq \lambda_A(n)$ et $n \notin I_A$*
  - *Si $m \vdash_A n$ avec $\lambda_A^{QA} = A$ alors $\lambda_A^{QA}(n) = Q$*

C'est la notion d'arène qui donne tout son intérêt à celle de chaîne pointée, et donc également à celle d'arbre pointé: ceux-ci permettent d'exprimer des parties entre Joueur et Opposant. On va maintenant formaliser les notions de partie et de partie légale, qui servent de base à la définition formelle de la notion de stratégie.

Definition 20. ***(partie avec connecteurs)*** *Une* ***partie avec connecteurs*** *dans une arène A est un arbre pointé s sur l'alphabet M_A et l'ensemble C de connecteurs qui satisfait:*

- *OP-*alternance*: si $x, y \in s$ tels que x précède juste (sans tenir compte d'un possible embranchement intermédiaire) y, alors $\lambda_A^{OP}(x) \neq \lambda_A^{OP}(y)$*
- *Un embranchement précède toujours un coup de l'Opposant.*
- *Si x pointe vers y alors $y \vdash_A x$.*

Definition 21. ***(partie légale avec connecteurs)*** *Une* ***partie légale avec connecteurs*** *est une partie où $\forall x \in s$, si x n'a pas de pointeur alors $x \in I_A$. L'ensemble des parties légales avec connecteurs sur A est noté $\mathcal{L}_A^+$.*

Definition 22. ***(domaine)*** *Soit s une partie légale avec connecteurs. On appelle le* ***domaine*** *de s (noté dom(s)) l'ensemble des parties sur lesquelles s a de l'information, c'est à dire l'ensemble des arbres préfixes s' de s dont chaque branche est un préfixe strict de la branche correspondante de s, et dont au moins une branche se termine par un coup de l'Opposant.*

La principale différence entre la sémantique de jeux et la théorie des jeux usuelle, c'est qu'on a besoin ici de parler de composition entre stratégie, donc d'exponentiation et d'interaction. Tel est l'intéret des définitions qui suivent : elles sont nécessaires pour définir la composition.

Definition 23. ***(arène vide)*** *On définit l'arène vide $\mathbf{1} =< \emptyset, \emptyset, \emptyset, \emptyset >$. Aucun coup n'y est possible, on a donc $\mathcal{L}_{\mathbf{1}}^+ = \{\epsilon\}$.*

Definition 24. ***(exponentiation)*** *Soient A et B deux arènes. On définit l'arène $A \Rightarrow B$ par:*

- $M_{A\Rightarrow B} = M_A \times M_B$
- $\lambda_{A\Rightarrow B} = [\lambda_A^{PO}, \lambda_B]$
- $I_{A\Rightarrow B} = I_B$
- $\vdash_{A\Rightarrow B} = (\vdash_A + \vdash_B) \cup \{(inr(b), inl(a) \mid b \in I_B \wedge a \in I_A\}$

On peut décrire l'exponentiation $A \Rightarrow B$ de la façon suivante : Les coups possibles sont ceux de A *et* de B, cependant la polarité Joueur/Opposant est inversée pour A, pour exprimer la contravariance de l'exponentiation, c'est-à-dire le rôle dual de l'Opposant.

4.1.3 Stratégies avec connecteurs

Definition 25. ***(stratégie avec connecteurs)*** *Une* ***stratégie avec connecteurs*** *σ pour une arène A (qu'on note $\sigma : A$) est un ensemble non-vide de* parties légales avec connecteurs *vérifiant les propriétés:*

- informativité*: si $s \in \sigma$, alors au moins une des branches de s se termine par un coup de P*
- accessibilité*: Si $s \in \sigma$, alors il existe une suite croissante atomique (*ie. *elle croît à chaque étape d'*un seul *mouvement de P, soit deux mouvements par alternance OP) d'éléments de σ préfixes de s qui mène de ϵ à s.*

Remarque: Dans le cas sans connecteurs, on a une condition de cloture par préfixe plutot que d'accessibilité, condition qui se révèle trop restrictive pour pouvoir exprimer complètement l'expressivité des stratégies avec connecteurs. Heureusement, ces deux conditions coïncident dans le cas sans connecteurs, lorsque les parties sont des chaînes pointées.

Definition 26. ***(déterminisme)***

- *Une stratégie avec connecteurs σ est dite* ***faiblement déterministe*** *si et seulement si, $\forall s, t \in \sigma$, toutes les branches du plus grand arbre préfixe commun à $dom(s)$ et $dom(t)$ finissent par un coup de P.*
- *Une stratégie avec connecteurs σ est dite* ***fortement déterministe*** *si et seulement si l'ordre préfixe est total sur tout $dom(s) \cap \sigma$.*

Dans la suite, sauf mention contraire, toutes les stratégies avec connecteurs seront supposées faiblement déterministes.

4.2 Interaction entre stratégies avec connecteurs

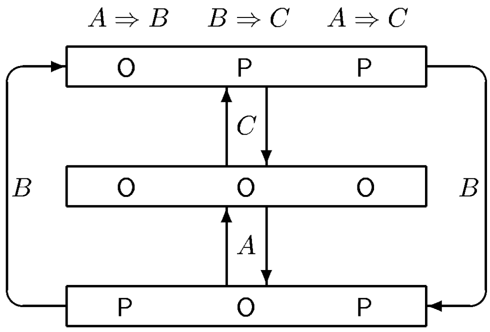
Definition 27. ***(interaction legale avec connecteurs)*** *Soient A, B, C des arènes. Une* ***interaction legale avec connecteurs*** *pour A, B et C est un arbre pointé u sur l'alphabet $M_A + M_B + M_C$, vérifiant que tous ses chemins appartiennent à $I(A, B, C)$ On notera $I^+(A, B, C)$ l'ensemble de toutes les interactions légales avec connecteurs sur A, B et C.*

Remarque: Lors de l'interaction entre σ et τ, il est nécessaire que σ ne puisse pas voir les embranchements générés par τ, et inversement, d'où la nécessité d'une *séparation* des interactions, qui permet aux stratégies de répondre à toutes les branches de la stratégie adverse, sans pour autant en distinguer les embranchements.

Lemme 5. ***(identification des embranchements)*** *Soit u une interaction avec connecteurs sur A, B et C. Alors on peut identifier dans u les embranchements générés par sa première composante $A \Rightarrow B$ (étiquetés "1") et ceux générés par sa seconde composante $B \Rightarrow C$ (étiquetés "2").*

Proof. Il suffit pour celà de suivre le parcours correspondant à l'interaction dans le diagramme ci-dessous. Plus précisément, on examine le coup juste avant l'embranchement:

- Si c'est un coup dans C, l'embranchement est étiqueté "2"
- Si c'est un coup dans A, l'embranchement est étiqueté "1"
- Si c'est un coup dans B, on regarde la position dans le diagramme.
  - Si c'est la case du dessus, l'embranchement est étiqueté "2"
  - Si c'est la case du dessous, l'embranchement est étiqueté "1"



□

Definition 28. ***(restriction avec séparation)*** *Soit u une interaction avec connecteurs sur A, B et C. Les trois restrictions suivantes sont obtenues de manière similaire: on sélectionne un sous-ensemble des embranchements de u, et on* sépare *chacuns d'entre eux : on génère autant de copies de u qu'il y avait de branches à cet embranchement, chacune avec une des branches possibles. On applique ensuite une restriction usuelle au sous-ensemble désiré à tous les arbres obtenus.*

- $u_{\upharpoonright^+_{A,B}}$ *est obtenu en séparant les embranchements de u étiquetés "2" et en appliquant la restriction usuelle $\upharpoonright_{A,B}$*
- $u_{\upharpoonright^+_{B,C}}$ *est obtenu en séparant les embranchements de u étiquetés "1" et en appliquant la restriction usuelle $\upharpoonright_{B,C}$*

Remarque: Dans ces deux restrictions avec séparation on cache une section externe de l'arène, supprimant de ce fait un des Joueurs de l'arène. Les branches que ce Joueurs avait généré doivent donc être séparées, car elles ne correspondent plus à l'action d'aucun Joueur mais à différentes parties distinctes répondant à plusieurs des choix non déterministes possibles de l'Opposant externe. Cependant si on cache une partie interne de l'arène, on aura pas de séparation : les embranchements corresponderont toujours à un des Joueurs. Par conséquent, la séparation avec restriction fait seulement sens pour cacher des parties externes de l'arène et la restriction habituelle fait seulement sens pour cacher des parties internes.

Definition 29. ***(composition)*** *Soient $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies; on définit tout d'abord leur composition parallèle:*

$$\sigma \| \tau = \{u \in I^+(A, B, C) \quad | \quad u_{\upharpoonright^+_{A,B}} \subseteq \sigma \quad \wedge \quad u_{\upharpoonright^+_{B,C}} \subseteq \tau\}$$

puis, comme habituellement, on cache la partie centrale de l'information:

$$\sigma;\tau = \{u_{\upharpoonright_{A,C}} \quad | \quad u \in \sigma\|\tau\}$$

*finalement, on ignore (enlève) les arbres non informatifs (*ie. *dont toutes les branches finissent par un coup de O) de l'ensemble obtenu par cette définition.*

Proposition 7. ***(composition)****Si σ et τ sont deux stratégies non déterministes dans $A \Rightarrow B$ et $B \Rightarrow C$ respectivement, alors $\sigma;\tau$ est une stratégie non déterministe pour $A \Rightarrow C$.*

Proof. L'informativité est conservée par définition. La propriété à vérifier est l'accessibilité : soit $s \in \sigma;\tau$, et soit u son témoin dans $\sigma\|\tau$. Par accessibilité sur σ et τ il existe des suites atomiques de préfixes $(v_i)_{i\in\{1\ldots k\}}$ et $(v'_i)_{i\in\{1\ldots k'\}}$ telles que $v_1 = v'_1 = \epsilon$, $v_k = u_{\upharpoonright_{A,B}}$, $v_{k'} = u_{\upharpoonright_{B,C}}$ et $\forall i, v_i \in \sigma$ et $v'_i \in \tau$. On remarque maintenant que $(v_i)_{i\in\{1\ldots k\}}$ et $(v'_i)_{i\in\{1\ldots k'\}}$ suffisent à construire u, puisqu'elles croissent de façon atomique et que chaque v_i (respectivement v'_i) correspond à un préfixe de $u_{\upharpoonright_{A,B}}$ (resp.$u_{\upharpoonright_{B,C}}$). Cette interaction entre les $(v_i)_{i\in\{1\ldots k\}}$ et les $(v'_i)_{i\in\{1\ldots k'\}}$ donne donc une suite atomique de préfixes de u, qui après restriction sur A, C montre bien l'accessibilité de s. □

Remarque: On a pas, en général, unicité du témoin dans la composition puisque dans le cas de stratégies seulement faiblement déterministes, les branches peuvent croitre dans des ordres différents.

Proposition 8. ***(conservation du déterminisme fort)*** *Soient σ et τ deux stratégies fortement déterministes, alors $\sigma;\tau$ est fortement déterministe.*

Proof. Soient $\sigma : A \Rightarrow B$ et $\tau : B \Rightarrow C$ deux stratégies fortement déterministes, alors $\sigma\|\tau$ vérifie également la condition de fort déterminisme, c'est à dire que $\forall u \in \sigma;\tau$, l'ordre préfixe est total dans $dom(u) \cap (\sigma\|\tau)$. En effet, si $u_1, u_2 \in \sigma\|\tau$ deux arbres préfixes d'un même u ne sont pas comparables par ordre préfixe, alors soit $u_{1\upharpoonright_{A,B}}$ et $u_{2\upharpoonright_{A,B}}$ soit $u_{1\upharpoonright_{B,C}}$ et $u_{2\upharpoonright_{B,C}}$ seraient non comparables, ce qui est absurde par fort déterminisme de σ et τ. On conclut par le fait que la condition de fort déterminisme passe naturellement de $\sigma\|\tau$ à $\sigma;\tau$ par restriction. □

Lemme 6. ***(décomposition d'une stratégie avec connecteurs)*** *D'une stratégie avec connecteurs σ on peut extraire des stratégies sans connecteurs $\mathcal{D}(\sigma) = (\sigma_i)_{i\in I}$ telles que tout chemin dans une interaction avec σ soit une interaction avec un σ_i. On définit alors, pour $s \in \sigma$, sa* ***composante en*** *σ_i comme le plus grand chemin s' de s tel que $s' \in \sigma_i$. On note que toute partie $s \in \sigma_i$ est la composante en σ_i d'une partie avec connecteurs $t \in \sigma$.*
Cette construction commute avec la composition, c'est à dire que si σ se décompose en $(\sigma_i)_{i\in I}$ et τ se décompose en $(\tau_j)_{j\in J}$, alors $\sigma;\tau$ se décompose en $\bigcup_{(i,j)\in I\times J} \sigma_i;\tau_j$

Proof. Soit σ une stratégie. On rappelle que pour un embranchement n-aire donnée, on peut *séparer* l'arbre correspondant à la stratégie, pour obtenir n arbres correspondants aux n choix possible pour l'embranchement. La décomposition consiste juste à séparer tous les embranchements dûs aux connecteurs. On obtient bien des stratégies au sens usuel:

- on ne garde que les chaînes de longueur paire
- supposons $s \in \sigma_i$. Il existe alors $t \in \sigma$ tel que s est la composante en σ_i de t, puisque s a nécessairement été obtenue à partir d'un tel t. Par accessibilité, on a alors une suite croissante atomique allant de ϵ à t. Sa composante dans σ_i ne peut croitre que d'un seul mouvement de P à la fois, ce qui donne la cloture par préfixe de s.
- σ est supposée faiblement déterministe. Soient $s_1, s_2 \in \sigma_i$. Par définition de σ_i il existe $s'_1, s'_2 \in \sigma$ telles que s_1 (resp. s_2) est la composante en σ_i de s'_1 (resp. s'_2). Par faible déterminisme, toutes les branches de $dom(s'_1) \wedge dom(s'_2)$ terminent par des coups de P, et $s_1 \wedge s_2$ est l'une d'elles donc termine par un coup de P.

Un chemin dans une interaction avec σ contient un choix pour tous les embranchements rencontrés, et est donc nécessairement compatible avec la composante de σ qui correspond à ces mêmes choix.
De plus cette construction commute effectivement avec la composition : pour tout chemin u de $\sigma \| \tau$ il existe i et j tels que $u \in \sigma_i \| \tau_j$. De même si $u \in \sigma_i \| \tau_j$, c'est forcément un chemin possible dans $\sigma \| \tau$. On conlut en restreignant l'alphabet des deux côtés. $\square$

Proposition 9. *Soit σ une stratégie avec connecteurs, soit $(\sigma_i)_{i \in I}$ sa décomposition, alors σ est faiblement déterministe si $\forall i \in I, \sigma_i$ est déterministe.*

Proof. Le sens $\Rightarrow$ a déjà été traité dans la preuve ci-dessus. Supposons que $\forall i \in I, \sigma_i$ est déterministe. Soient $s_1, s_2 \in \sigma$, soit t un chemin allant jusqu'à une feuille de $dom(s_1) \cap dom(s_2)$, mais terminant par un coup de O. Alors nécessairement il existe i tel que t est la composante en σ_i de $dom(s_1) \cap dom(s_2)$ (il suffit de noter les choix faits à chaque embranchement de $dom(s_1) \cap dom(s_2)$ pour obtenir le chemin t dans l'arbre). Alors soient s'_1 la composante en σ_i de s_1 et s'_2 la composante en σ_i de s_2. Alors $s'_1 \wedge s'_2 = t$ et termine par O, ce qui est absurde par déterminisme de σ_i. $\square$

Proposition 10. ***(conservation du déterminisme faible)*** *Soient σ et τ deux stratégies faiblement déterministes, alors $\sigma; \tau$ est faiblement déterministe.*

Proof. Le déterminisme faible ne dépend que du déterminisme des composantes, or ce déterminisme est stable par composition. $\square$

Remarque: On désire maintenant prouver l'associativité, c'est à dire que $\sigma; (\tau; \sigma) = (\sigma; \tau); \delta$. Pour ce faire, on va définir une notion d'interaction à quatre arènes.

Definition 30. *On étend la notion d'interaction légale à quatre arènes A, B, C, D. L'ensemble $I^+(A, B, C, D)$ est défini comme l'ensemble de toutes les parties légales s avec connecteurs dont tous les chemins sont dans $I(A, B, C, D)$. Cette définition s'étend directement à un nombre fini quelconque d'arènes.*

Lemme 7. ***(zipping)*** *Soit $u \in I^+(A, C, D)$, supposons $\forall t \in u_{\upharpoonright^+_{A,C}} \exists v_t \in I^+(A, B, C) t = v_{t \upharpoonright_{AC}}$ alors $\exists w \in I^+(A, B, C, D)$ tel que $w_{\upharpoonright_{A,C,D}} = u$ et $w_{\upharpoonright^+_{A,B,C}} = \{v_t | t \in u_{\upharpoonright^+_{AC}}\}$*

Proof. On raisonne par induction sur $\Sigma_{t \in u_{\upharpoonright^+_{AC}}} |t|$.

- Si $\Sigma_{t \in u_{\upharpoonright^+_{AC}}} |t| = 0$,
  alors u ne joue que dans D et on peut poser $w = u$. On vérifie bien que $u_{\upharpoonright_{ACD}} = u$ et $u_{\upharpoonright^+_{ABC}} = \{v_t | t \in u_{\upharpoonright^+_{AC}}\}$ (ces deux ensembles étant vides, puisque u ne joue que dans D).
- Sinon soit $t \in u_{\upharpoonright^+_{AC}}$, et soit a un dernier coup de t. On applique l'hypothèse d'induction à $u' = u_{<a}$ (le plus grand arbre préfixe de u ne contenant pas a) et à $\{(v_t)_{<a} | t \in u_{\upharpoonright^+_{AC}}\}$, qui vérifient bien que $t_{<a} = (v_t)_{<a \upharpoonright_{A,C}}$. On obtient alors un $w' \in I^+(A, B, C, D)$ tel que $u' = w'_{\upharpoonright_{ACD}}$ et $w'_{\upharpoonright^+_{ABC}} = \{(v_t)_{<a} | t \in u_{\upharpoonright^+_{AC}}\}$.
  On va maintenant compléter ce w' de la façon suivante : a est un coup dans $A \Rightarrow C$ et est un dernier coup dans $u_{\upharpoonright^+_{AC}}$, donc les éventuels coups qui le suivent dans t sont soit tous dans B, soit tous dans D. Par conséquent a est soit un dernier coup de u, soit un dernier coup de v_t. On peut donc étendre la branche de w' correspondant à t ou v_t (comme convient), d'où $w \in I^+(A, B, C, D)$, qui vérifie donc que $w_{\upharpoonright_{ABC}} = u$ et $w_{\upharpoonright^+_{ABC}} = \{v_t | t \in u_{\upharpoonright^+_{AC}}\}$.

$\square$

Proposition 11. ***(associativité)*** *Soient $\sigma : A \Rightarrow B$, $\tau : B \Rightarrow C$ et $\delta : C \Rightarrow D$; alors*

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

Proof. Soit $s \in (\sigma;\tau);\delta$, s a un témoin $u \in I^+(A,C,D)$, qui vérifie que $\forall t \in u_{\upharpoonright^+_{AC}}$, t a un témoin v_t dans $I^+(A,B,C)$, vérifiant donc que $t = v_{t\upharpoonright_{AC}}$. Le lemme de zipping nous donne donc le témoin w dans $I^+(A,B,C,D)$. Dès lors on vérifie :

- $w_{\upharpoonright^+_{AB}} = (w_{\upharpoonright^+_{ABC}})_{\upharpoonright^+_{AB}} = \bigcup_{t \in u_{\upharpoonright^+_{AC}}} v_{t\upharpoonright^+_{AB}} \subseteq \sigma$
- Soit $s' \in w_{\upharpoonright^+_{BD}} = (w_{\upharpoonright^+_{BCD}})_{\upharpoonright_{BD}}$ donc $\exists v' \in w_{\upharpoonright^+_{BCD}} v'_{\upharpoonright_{BD}} = s'$, et on vérifie de plus que :
  - $v'_{\upharpoonright^+_{BC}} \subseteq (w_{\upharpoonright^+_{BCD}})_{\upharpoonright^+_{BC}} = (w_{\upharpoonright^+_{ABC}})_{\upharpoonright^+_{BC}} \subseteq \bigcup_{t \in u_{\upharpoonright^+_{AC}}} (v_t)_{\upharpoonright^+_{BC}} \subseteq \tau$
  - $v'_{\upharpoonright^+_{CD}} \subseteq (w_{\upharpoonright^+_{BCD}})_{\upharpoonright^+_{CD}} = (w_{\upharpoonright_{ACD}})_{\upharpoonright^+_{CD}} \subseteq u_{\upharpoonright^+_{CD}} \subseteq \delta$

Ce qui conclut la preuve : $w_{\upharpoonright_{ABD}}$ est donc un témoin pour u dans $\sigma;(\tau;\delta)$. Le sens inverse s'en déduit par miroir. □

Definition 31. ***(stratégies séquentielles)*** *Une stratégie avec connecteurs est dite* ***séquentielle*** *si elle n'admet pas de sous-suite atomique d'arbres préfixes qui joue dans une branche a, dans une autre branche b, puis revient sur a. Toutes les branches sont visitées de façon séquentielle.*

4.2.1 Innocence dans les stratégies avec connecteurs

Definition 32. ***(innocence)*** *Une stratégie avec connecteurs est* ***innocente*** *si et seulement si toutes ses composantes sont innocentes.*

Remarque: C'est bien la bonne notion d'innocence sur les stratégies avec connecteurs. En effet l'innocence sur les stratégies avec connecteurs doit coïncider avec l'innocence usuelle dans le cas sans connecteurs, de plus on voit mal comment une P-vue pourrait sortir de la composante actuelle.

Proposition 12. ***(conservation de l'innocence)*** *Soient σ et τ deux stratégies innocentes, alors $\sigma;\tau$ est innocente.*

Proof. L'innocence ne dépend que de l'innocence des composantes, et est préservée par composition dans le cas des stratégies classiques. Elle est donc préservée par composition dans le cas des stratégies avec connecteurs. □

4.3 Décomposition de PCF_1

4.3.1 Modèle séquentiel de PCF_1

La sémantique de PCF_1 en termes de stratégies avec connecteurs est comparable à celle du modèle minimal, sans réponse aux questions; on ne se préoccupe pas du parenthèsage. Toutes les stratégies considérées sont innocentes. Initialement, on interprète le $\wedge$ comme un connecteur opaque, ce qui correspond à un embranchement. Ensuite, on lui adjoint la règle de convergence suivante : une interaction converge si et seulement si toutes ses branches convergent, ce qui équivaut à dire qu'on ne rencontre pas $\perp$ durant l'exécution. Si Σ désigne l'arène correspondant au type de base, les stratégies sur Σ correspondent aux formes normales de type o dans lesquelles on aurait pas encore résolu la sémantique opérationnelle du $\wedge$. Ce critère de convergence revient précisément à ramener ces stratégies au cas convergence/divergence.

Definition 33. ***(Catégorie des stratégies séquentielles unaires avec $\wedge$)*** *L'ensemble des stratégies avec connecteurs décrites plus haut est une catégorie, on la note* ***Seq****.*

Remarque. Ici le terme de catégorie n'est pas crucial pour la compréhension, mais nécessaire pour la démonstration, car on aura besoin de la notion de foncteurs pour prouver la pleine adéquation de ce modèle pour PCF_1. Cependant le lecteur sans connaissance du sujet ne devrait pas se trouver trop handicapé.

Proposition 13. ***(foncteurs de déséquentialisation)*** *Soit* ***Gam*** *le modèle de jeux usuel de PCF_1 (avec les réponses et le parenthésage). Alors il existe un foncteur bijectif $\mathcal{F}$:* **Gam** $\rightarrow$ **Seq**

Proof. L'image des objets par $\mathcal{F}$ est triviale puisque la notion d'arène est la même dans les deux cas. Soit une stratégie $\sigma : A \Rightarrow B$ dans **Gam**. A chaque partie dans σ on associe une partie avec connecteurs de la façon suivante : chaque fois qu'une nouvelle question suit une réponse, on la place dans une autre branche qui vient se greffer là. On obtient un arbre où dès qu'on rencontre une réponse, on ne peut obtenir que d'autres réponses, et de plus par construction du modèle ces réponses ne peuvent être étiquetée que t; elles n'apportent donc pas plus d'informations et on peut les oublier. Cette contruction est évidemment bijective, puisqu'il ne s'agit que d'une représentation différente sans aucune perte d'information. Il suffit pour retrouver la stratégie originelle de refermer toutes les parenthèses, puis, pour chaque embranchement, d'insérer la branche de droite (bien parenthèsée) au niveau de l'embranchement.

Pour avoir la fonctorialité, il faut également noter qu'on peut faire subir la même transformation à une interaction : la structure d'arbre (aux branches parenthèsées trivialement) est induite par les parenthèses. On obtient bien une interaction dans **Seq**, puisque toute P-vue d'une branche dans l'interaction obtenue est la P-vue d'une branche de l'interaction originelle, et inversement. □

Corollaire 1. *Cette bijection assure qu'on a une correspondance directe entre les termes de PCF_1 et les stratégies dans* ***Seq****.*

4.3.2 Modèle clos par préfixe de PCF_1

Si σ est une stratégie dans **Seq**, on note $\overline{\sigma}$ sa cloture par préfixe. De même, on note $\overline{\mathbf{Seq}}$ l'ensemble de ces stratégies. Soit $\tau \in \overline{\mathbf{Seq}}$, alors il y a plusieurs façons de retrouver un $\sigma \in \mathbf{Seq}$ tel que $\tau = \overline{\sigma}$, il suffit de trier les parties de τ de façon cohérente avec l'ordre partiel préfixe pour obtenir une stratégie séquentielle.

Proposition 14. *Soient $\sigma, \tau \in$* **Seq***, alors $\overline{\sigma} = \overline{\tau} \Rightarrow \sigma \equiv \tau$.*

Proof. Supposons que σ et τ jouent sur l'arène A. Soit $\alpha \in \mathbf{Seq}$ une stratégie sur l'arène $A \Rightarrow \Sigma$. Considérons l'interaction $\sigma \| \alpha$. Elle n'est pas nécessairement identique à $\overline{\sigma} \| \overline{\alpha}$, cependant il existe dans ces deux ensembles un élément maximal, et c'est nécessairement le même, car σ (resp. α) et $\overline{\sigma}$ (resp. $\overline{\alpha}$) ne diffèrent que par leur ordre associé, et l'ordre de l'un est une extension de l'ordre de l'autre. Le même raisonnement est valable pour l'interaction $\tau \| \overline{\alpha}$. On conclut en disant que le résultat de la composition ne dépend que de cet élément maximal dans l'interaction, puisque après application des règles de convergence liées au modèle (c'est à dire qu'une partie sur l'arène Σ converge si et seulement si toutes ses branches convergent), la stratégie obtenue est dans l'arène Σ qui ne contient que les constantes $\mathtt{tt}$ et $\perp$ et donc ne contient pas d'information sur un ordre partiel. □

On en conclut que $\overline{\mathbf{Seq}}$ est bien un modèle de $PCF_1/\equiv$ puisqu'il n'égalise que des stratégies contextuellement équivalentes.

4.3.3 Modèle décomposé de PCF_1

Si σ est une stratégie dans $\overline{\mathbf{Seq}}$, soit $\mathcal{D}(\sigma)$ sa décomposition (comme présenté au lemme 6). Soit $(\tau_i)_{i \in I} \in \mathcal{D}(\overline{\mathbf{Seq}})$, il y a plusieurs façons de retrouver un $\sigma \in \overline{\mathbf{Seq}}$ tel que $\tau = \mathcal{D}(\sigma)$, il suffit de regrouper les stratégies de τ à n'importe quel point de leur préfixe commun.

Proposition 15. *Soient $\sigma, \tau \in \overline{\mathbf{Seq}}$, alors $\mathcal{D}(\sigma) = \mathcal{D}(\tau) \Rightarrow \sigma \equiv \tau$.*

Proof. Supposons que σ et τ jouent sur l'arène A. Soit $\alpha \in \overline{\mathbf{Seq}}$ une stratégie sur l'arène $A \Rightarrow \Sigma$. Considérons encore une fois l'interaction $\sigma \| \alpha$. De la même façon que précédemment, sa restriction à Σ, après application des critères de convergence, ne peut contenir aucune information sur la localisation des branchements dans l'interaction, et elle doit donc être égale à la restriction à Σ de $\tau \| \alpha$. □

4.3.4 Corollaires de la décomposition

Cette décomposition permet de voir que jouer contre une stratégie dans **Seq** revient à jouer indépendemment contre toutes ses composantes.

Corollaire 2. *L'égalité dans $\Lambda / \equiv_{PCF_1}$ est décidable.*

Proof. C'est une conséquence de la décomposition : deux termes de Λ sont séparables par un terme de Λ si et seulement si ils le sont par un terme de PCF_1. □

Corollaire 3. *L'égalité est décidable dans $PCF_1 / \equiv_{PCF_1}$ et dans $PCF_1 / \equiv_\Lambda$.*

Proof. La décomposition nous permet de ramener les deux cas à $PCF_1 / \equiv_\Lambda$. Elle nous permet aussi de cellulariser tout terme de PCF_1, en cellularisant toutes ses composantes. De la même façon elle nous donne un algorithme de listing : pour un type donné on construit toutes les coordinations possibles de toutes les classes d'équivalences de Λ. □

Ces nouvelles preuves, ainsi qu'un raisonnement simple sur les propriétés du tableau, permettent de résumer ainsi la nouvelle situation :

	Λ	PCF_1	$PCF_{1.5}$	PCF_2	PCF
Λ	$+$	$+$			
PCF_1	$+$	$+$			
$PCF_{1.5}$					
PCF_2				$-$	$-$
PCF					$-$

Le cas $PCF_2 / \equiv_{PCF}$ peut sans doute être montré indécidable en adaptant la preuve de R.Loader pour $PCF_2 / \equiv_{PCF_2}$, mais aucun argument simple sur la structure du tableau ne permet de le conclure. On remarque que les lignes et colonnes contenant $PCF_{1.5}$ restent désespérément vierges. En effet, aucune des méthodes n'existant auparavant ou développées ici ne permettent de l'attaquer. La cellularité ne semble pas s'appliquer, le flot de contrôle devenant *dynamique*, et d'un autre côté il n'y a pas la symétrie `tt`/`f` qui permet à Loader de montrer l'indécidabilité de PCF_2.

5 Conclusion

Finalement, on a développé des outils qui permettent de mieux comprendre le rôle des connecteurs, c'est à dire de structures de contrôle locales. Ces outils ont permis d'étendre la preuve de décidabilité de Λ à PCF_1, par la remarque que le modèle de jeux de PCF_1 est finalement identique à celui de *Lambda*, aux branchements près. Cependant, et même avec la vision que procure les stratégies avec connecteurs, le cas de sesqui-PCF reste inabordable. D'une part la présence du `ff` bloque les méthodes impliquant la cellularité utilisées dans Λ et d'autre part le manque de symétrie de la sémantique opérationelle par rapport à `tt` et à `ff` empèche de généraliser l'indécidabilité de PCF_2 à $PCF_{1.5}$. Pourtant ces constatations laissent entrevoir plusieurs voies de recherche :

Tout d'abord, la notion de cellularité mérite qu'on s'y attarde. Non seulement les termes cellulaires de Λ permettent de décider l'équivalence, mais ils possèdent en dehors de celà des propriétés étonnantes, en ce sens que l'interaction entre eux est grandement simplifiée. Etrangement, leur formulation naturelle dans les jeux dépasse la syntaxe usuelle des λ-termes, puisqu'elle romp l'innocence. Une des questions serait alors de rechercher à partir de cette sémantique la syntaxe naturelle pour les stratégies cellulaires, et de faire le point sur leur expressivité.

Une autre voie de recherche part d'une constatation : on peut encoder toutes les versions imaginable de PCF dans Λ, en prenant pour les constantes les encodages habituels du λ-calcul (par exemple en prenant $\lambda xy.x$ pour `tt` et $\lambda xy.y$ pour `ff`). Dès lors l'équivalence contextuelle y devient décidable mais ce n'est pas la même, l'environnement admettant entre autre l'encodage de `call/cc`, parfaitement correct dans le modèle sans parenthèses de Λ. On constate alors que rajouter le contrôle dans un langage a beau augmenter sa complexité, elle tend à y simplifier le problème de l'équivalence contextuelle. On peut donc tenter d'analyser les procédures de contrôle non intuitionnistes comme `call/cc`, ou le `shift/reset` d'H.Herbelin[2], et de regarder ce qu'elles donnent du point de vue sémantique des jeux et décidabilité, pour peut-être plus tard revenir sur le problème initial de $PCF_{1.5}$.

References

[1] Jean-Yves Girard. Geometry of interaction. i. interpretation of system f. *LICS*, 1988.

[2] A.Sabry H. Herbelin, Z.Ariola. A type-theoretic foundation of delimited continuations. 2004.

[3] Russ Harmer. Innocent game semantics.

[4] Ralph Loader. An algorithm for the minimal model. Unpublished.

[5] Ralph Loader. Unary pcf is decidable. *Theorical Computer Science*, 1998.

[6] Ralph Loader. Finitary pcf is undecidable. *Theorical Computer Science*, 2001.

[7] L. Hong M. Hyland. On full abstraction for pcf.

[8] Vincent Padovani. Decidability of all minimal model - revised version. Unpublished.

[9] Vincent Padovani. *Filtrage d'ordre superieur.* PhD thesis, 1995.

[10] Manfred Shmidt-Shauss. Decidability of behavioural equivalence in unary pcf. *Theorical Computer Science*, 1999.