École Normale Supérieure de Lyon
Magistère Informatique et Modélisation
(Lyon)

INRIA
(Sophia Antipolis)

Modèles du Π-Calcul. Structures d'Actions et Relations avec l'Équivalence Par Permutations.

mémoire de D.E.A.
Année 1993-1994

Dal-Zilio Silvano.

“ Les lettres fondamentales sont au
nombre de vingt-deux, et avec celles-là
Dieu forma toute la création.”
in Sefer Jesirah

Remerciements.

Je tiens à remercier vivement toutes les personnes qui ont contribué à rendre mon stage si enrichissant. Et tout spécialement **Gérard Boudol** et **Ilaria Castellani** pour leur encadrement et leur conseils qui ont permis de développer mon intérêt pour la sémantique et la recherche en général.

Chapitre 1

Π-calcul.

Le but de notre étude est d'explorer deux modèles différents du π-calcul. Le premier, qui utilise les structures d'actions, un nouveau type d'algèbre destiné à l'étude des modèles du calcul concurrent, a été longuement développé par Milner. Il est centré sur les notions d'effet et d'action. Le deuxième modèle proposé, est une approche originale basée sur la notion d'équivalence par permutation de séquences de transitions. Cette technique, développée par Berry et Lévy et reprise avec succès par Boudol et Castellani pour CCS [6][7][8][9], utilise les preuves des transitions d'une “SOS” comme étiquette : on parle de transitions prouvées. Elle fournit une sémantique dans laquelle tous calcul est représenté par un *pomset*[1].

Nous commencerons par un bref aperçu du π-calcul, suivi par la description des structures d'actions. Nous essayerons d'en présenter les aspects fondamentaux et nous introduirons comme exemple PIC, une structure d'action pour le π-calcul proposée par Milner. Le chapitre suivant sera consacré à la définition d'une équivalence par permutations pour deux systèmes de transitions, EPIC et LPIC, qui représentent des sémantiques opérationnelle du π-calcul. L'introduction de ces deux modèles sera l'occasion de mener une étude comparée, dont le but est de définir une nouvelle structures d'actions pour le π-calcul. Nous verrons en conclusion quel est le bilan de cette tentative.

Nous ne nous attarderons pas sur la définition du π-calcul, mais une brève introduction permettra de suivre nos tentatives de modélisation. Le π-calcul [15][17][18] est un calcul de processus basé sur la notion de “*nommage*”. Son développement est historiquement lié à la restriction de CCS, dont il réalise une remarquable généralisation et simplification, qui ne permettait pas d'utiliser les ports de communication comme valeurs. Bien que sa syntaxe soit très simple

[1] *i.e.* un multi-ensemble partiellement ordonné.

(voir tableau 1.1 [2]), il permet de décrire les systèmes concurrent dans lesquels les processus sont mobiles, *i.e.* dans lesquels la topologie des communication peut évoluer dynamiquement. De plus l'existence de modélisation du λ-calcul paresseux, aussi bien que du λ-calcul avec appel-par-valeur [11][5], ainsi que la définition de sémantique pour des languages objets [20] en fait un candidat privilégié pour la fondation de nouveaux langages de programmation parallèle.

L'interprétation intuitive des constructeurs du π-calcul est simple. On considère un ensemble infini de noms $x, y, \ldots$.

- $\bar{x}y$ émet le nom y sur le canal x. L'émission est bloquante, *i.e.* elle bloque, dans le processus $\bar{x}y.P$ par exemple, toutes réactions impliquant P.
- $x(y)$ lie le nom y dans le processus $x(y).P$ et attend qu'un nom soit émis sur le canal x pour le capturer.
- $\|$ est l'opérateur de composition parallèle entre termes. Deux processus en parallèle peuvent évoluer indépendamment ou se synchroniser lors d'une communication.
- $(\nu\ x)P$ déclare le nom x comme privé, *i.e.* seul le processus P peut émettre et recevoir sur ce canal.
- $!P$, ou bang P, permet de coder la récursion. Son comportement dans la sémantique n'est cependant pas très intéressant, et nous ne l'étudierons pas profondément dans la suite de notre étude.
- $\mathsf{0}$ représente le processus inerte. Il pourrait être remplacé par le terme $(\nu x)\ (\bar{x}x)$.

Inspirée par la machine chimique [3][4], la relation de réduction définie sur le π-calcul repose sur une congruence, $\equiv$, qui permet de ne pas se préoccuper de détails structuraux. Intuitivement, $\equiv$ permet de mettre en contact les participants d'une communication. On a par exemple :

$$\begin{array}{ll} (P \parallel Q) \parallel R \equiv P \parallel (Q \parallel R) & \\ (\nu\ x)P \parallel Q \equiv (\nu\ x)(P \parallel Q) & \quad \text{si } x \text{ n'est pas un nom libre de } Q \\ \ldots & \end{array}$$

la réduction $\rightarrow$, ne comporte qu'un seul axiome. Il modélise la communication. Le reste ne représente que des règles d'inférence qui permettent la communication à l'intérieur de certains termes.

[2] On utilise dans notre étude une version simplifiée du π-calcul monadique, proche du "mini-π-calcul" utilisé par Boudol dans [5], qui ne contient pas le constructeur somme, $+$. On montre que le pouvoir expressif reste inchangé

$$P ::= \quad 0 \quad | \quad \pi.P \quad | \quad P \parallel P \quad | \quad !\,P \quad | \quad (\nu\ x)\ P$$

$$\pi ::= \quad x(y) \quad | \quad \bar{x}y$$

FIG. 1.1 - syntaxe du π-calcul.

$$\text{Comm.} \qquad x(y).P \parallel \bar{x}z.Q \rightarrow P\{z/y\} \parallel Q$$

$$\text{Par.} \qquad \frac{P \rightarrow P'}{P \parallel Q \rightarrow P' \parallel Q}$$

$$\text{Res.} \qquad \frac{P \rightarrow P'}{(\nu\ x)P \rightarrow (\nu\ x)P'}$$

$$\text{Struct.} \qquad \frac{Q \equiv P \quad P \rightarrow P' \quad P' \equiv Q'}{Q \rightarrow Q'}$$

Malgré sa grande généralité, le π-calcul permet d'observer des phénomènes "très fins". Comme par exemple l'utilisation de ressources limitées, ou l'étude du "parallélisme" dans les réduction du λ-calcul. C'est pourquoi, afin de bien étudier tous ces mécanismes, plusieures tentatives de modélisation du π-calcul ont été faites. Nous nous proposons dans la suite de notre étude de vous présenter deux de ces tentatives.

Chapitre 2

Structures d'actions.

Les structures d'actions ont été introduites par Milner comme un outil d'étude des différents modèles concrets proposés pour l'étude de la concurrence, ainsi que des systèmes d'interactions en général. Cette variété "d'algèbre" est équipée de deux opérateurs, la composition (.) et le produit d'actions ($\otimes$), et comprend une famille indexée d'éléments, les *abstracteurs*, qui permettent la paramétrisation des actions. On définit également une relation, appelée *réaction*, afin de représenter l'activité.

Ce formalisme se veut suffisamment général et flexible pour permettre d'y coder le plus grand nombre de modèles du parallélisme. Par exemple Milner propose, dans [13][14], une représentation des calculs de processus par la superposition aux structures d'actions de "structures de processus" (comme `0`,+ ou `!`). Malgré sa généralité, la théorie des structures d'actions met l'accent sur deux notions : l'*atomicité* et les *effets* (*i.e.* l'influence exercée par la réalisation d'une action sur ses participants). La notion d'effet, associée à la notion d'*incident*[1] permettant en particulier un traitement uniforme de la congruence par bisimulation.

2.1 Définition.

Une structure d'actions $(M, X, A, \searrow)$ est définie par la donnée de:

– Un ensemble de noms : X

– $(M, \otimes, \epsilon)$, monoïde des arités.

– Pour tout $m, n \in M$, un ensemble $A_{m,n}$ d'actions. Ce sont les actions d'arité $m \to n$.

[1] un incident peut-être interprété comme un ensemble d'actions "observables"

– Un ensemble d'actions "neutres": $\{id_n / n \in M\}$, et d'opérateurs unaires d'abstraction : $\{ab_x / x \in X\}$.

– Deux opérateurs binaires. La composition ., et le produit tensoriel $\otimes$.

Ces différents opérateurs sont soumis à des règles d'arité[2]:

$$id_n : n \to n \qquad \frac{a : k \to n \qquad b : m \to p}{a \otimes b : k \otimes m \to n \otimes p}$$

$$\frac{a : k \to n \qquad b : n \to p}{a.b : k \to p} \qquad \frac{a : k \to n}{ab_x a : ab_x k \to ab_x n}$$

et ils vérifient huit axiomes (on suppose ne composer que les actions d'arité convenable):

$$\begin{array}{rlrl} C1 & a.id = a = id.a & C2 & a.(b.c) = (a.b).c \\ P1 & a \otimes id_\epsilon = a = id_\epsilon \otimes a & P2 & a \otimes (b \otimes c) = (a \otimes b) \otimes c \\ PF1 & id \otimes id = id & PF2 & (a.b) \otimes (c.d) = (a \otimes c).(b \otimes d) \\ AF1 & ab_x id = id & AF2 & ab_x(a.b) = (ab_x a).(ab_x b) \end{array}$$

Une structures d'actions est également munie d'une relation de réduction, ou *réaction*, sur les actions: $\searrow$. La réaction est un préordre tel que:

– $A_{m,n}$ est stable par $\searrow$, *i.e.* la réaction préserve l'arité.

– Les opérateurs préservent $\searrow$.

– id est inactive : $id \searrow a \Rightarrow a = id$.

On remarque que Milner a défini un modèle algébrique pour tenter de représenter l'ensemble des modèles du parallélisme. Ce choix s'explique par l'état actuel des travaux dans le domaine de la "vraie concurrence". En particulier il existe des calculs de processus algébriques et des théories algébrique pour les réseaux de Petri. Il s'explique également par l'absence d'une structure canonique qui expliciterait aussi bien la dynamique que la construction des processus. L'utilisation de domaines "à la Scott"[6][8][9], par exemple, est une voie de recherche fructueuse, qui permet d'appréhender globalement les phénomènes du parallélisme, mais ils ne possèdent pas les propriétés des structures d'actions, qui sont une algèbre pour, à la fois, la syntaxe et la sémantique des "calculs interactifs".

[2]on suppose avoir défini ab_x sur M pour la règle sur l'abstraction

Mais comme nous l'avons dit en introduction, l'approche choisie pour les structures d'actions n'est pas faite uniquement de généralités. Elle souligne deux notions considérées comme centrales : *Atomicité* et *effet*. L'atomicité : une action doit pouvoir être appréhendée du point de vue microscopique aussi bien que macroscopique[3]. Et les effets : contrairement aux calculs classiques qui mettent l'accent sur les transitions (CCS par exemple), ce sont les résultats des actions qui jouent le rôle principal. Ces notions sont naturelles si on veut voir les structures d'actions comme une tentative plus générale de comprendre ce qu'est une interaction.

Il suffit de choisir un exemple dans la vie courante pour s'en convaincre. Conduire une voiture est une succession d'actions complexe (accélération, dépassement,...), qui se résume par une unique action: "je roule". Cette action engage plusieurs participants (les conducteurs qui vous entourent), et ses effets les impliquent différemment.

Dans un domaine plus théorique, qui est celui du π-calcul, on peut considérer la transition:

$$(\nu\ x)(\bar{y}x.P) \parallel y(z).Q \longrightarrow (\nu\ x)(P \parallel Q\{x/z\})$$

les actions exercées par chaque sous-terme sont l'attente d'un message sur le canal y d'une part, et l'émission d'un nom privé sur ce même canal d'autre part. Ces deux actions entraînant une communication. Cette réaction peut s'interpréter sous forme de réduction sur les actions:

$$(\nu x).(\bar{y}x) \otimes y(z) \searrow (\nu x).(id \otimes \{x/z\})$$

de plus on peut observer les différents effets sur les termes. Le processus P ne subit aucun effet[4] (action id) tandis que Q est soumis à une substitution et que la restriction s'est déplacée.

Cet exemple permet de mieux comprendre les structures d'actions. Une réaction représente un pas dans la réalisation d'une action. Et ce sont les action inertes, *i.e.* incapables de réagir, qui représentent les effets. Au cours des différentes réactions qui peuvent s'enchaîner, les effets s'accumulent pour donner l'effet global d'une action :

$$a \searrow a_1.e_1 \ (e_1 \text{ effet}), \qquad a_1 \searrow a_2.e_2 \ (e_2 \text{ et } e_1.e_2 \text{ effets}), \qquad \ldots$$

Le cas du π-calcul est d'ailleurs intéressant, puisqu'il nous fournit des exemples dans lesquels l'effet d'une réaction sur un produit est plus important que le produit des réactions sur chaque partie (c'est le cas de l'extension de la portée de la restriction).

[3] *i.e.* comme la composition d'autres actions plus élémentaires

[4] il subit en fait un effet plus subtil, puisque l'élimination de la garde "libère" le processus

2.2 Effets et structures d'effets.

Comme nous venons de le voir, un effet est une action dont la particularité est d'être inerte.

`Définition (inertie).`

Une action a est dite inerte si, lorsqu'elle est postcomposée à une autre action, elle ne contribue à aucune réaction. i.e. *:*

$$\forall b.a \searrow c, \exists b' \quad \begin{array}{l} b \searrow b' \\ et\ c = b'.a \end{array}$$

En particulier une telle action est inactive ($\exists a', a \searrow a' \Rightarrow a' = a$).

Il est d'ailleurs intéressant de définir la structure que devrait posséder l'ensemble de tous les effets (on s'attend a trouver une sous-structure d'action). Quelques propriétés de cet ensemble sont naturelles : les effets doivent pouvoir être composés, et les actions id_n doivent former un ensemble particulier d'effets. Mais il faut également que tous les effets possibles d'une action soient ordonnés. Intuitivement, cette condition permet de respecter un principe de causalité. Supposons que l'action a se décompose en $a_1.b_1$ et $a_2.b_2$, avec b_1, b_2 effets. La relation qui ordonne les décompositions d'une action (en action composée à un effet) est définie par (voir figure 2.1):

$$(a_1, b_1) \leq (a_2, b_2) \quad \text{si } \exists b \text{ effet}, \quad a_1 = a_2.b$$

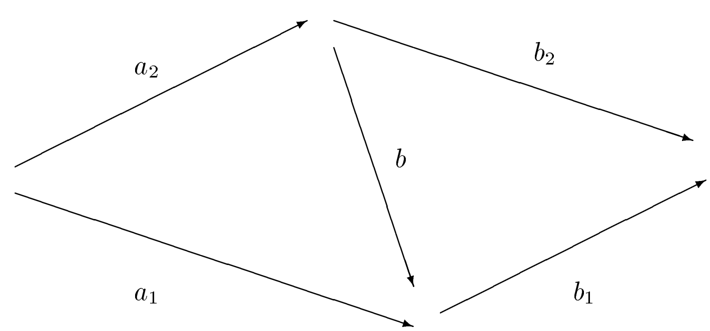


FIG. 2.1 - Ordre sur la décomposition d'une action.

`Définition (structure d'effets).`

Un sous-ensemble E de A est une structure d'effets si E est une sous-structures d'actions composée d'éléments inertes, et si pour chaque action a, l'ensemble de ses décompositions suivant E : $\mathcal{D}(a, E) = \{(a', e)/a = a'.e, e \in E\}$, est un `postcomposant` *de A,* i.e. *un ensemble dirigé par rapport à $\leq$ (voir fig.2.2):*

$$\begin{array}{lcll} a = a_1.e_1 = a_2.e_2 & & & \\ a_1, a_2 \in A & \Rightarrow & \exists a' \in A, & a_i = a'.e_i' \;\; et \;\; e_1'.e_1 = e_2'.e_2 \\ e_1, e_2 \in E & & \exists e_1', e_2' \in E & \end{array}$$

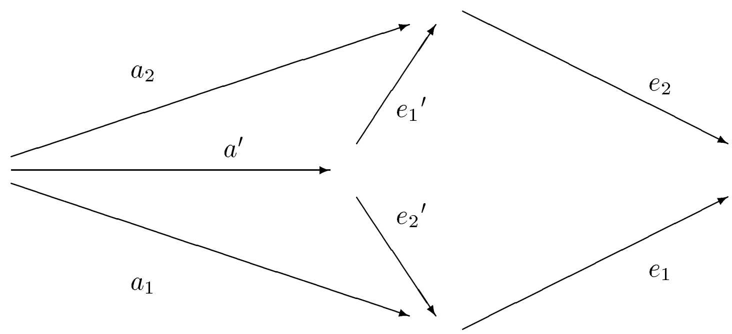


FIG. 2.2 - Décomposition dans un postcomposant.

2.3 Structures de processus.

Comme nous le laisse entrevoir notre exemple informel de la section 2.1, il est possible de modéliser par des actions l'ensemble des "transitions du π-calcul". Mais il est aussi possible de modéliser le π-calcul dans son entier. Une solution consiste à modéliser par une action aussi bien chaque transition que chaque processus. C'est ce que propose Milner avec ses "calculs d'actions"[5].

Une autre solution, que nous présentons ici, consiste à rajouter à la structures d'actions des transitions, un ensemble de constructeurs de processus. Cette démarche est d'ailleurs facilement généralisable aux différents calculs (de processus) existants.

`Définition (Structure de processus).`

Il est possible de définir à partir d'une structures d'actions $(M, X, A, \searrow_A)$ *quelconque, une structure de processus* $\mathcal{P}_A$*. Les processus étant définis par la syntaxe (*P *désigne un processus et* a *une action de* A*)*

[5] "action calculi",[16]

$$P ::= \quad a.P \quad \Big| \quad P \otimes P \quad \Big| \quad (x)P \quad (\simeq ab_x P) \quad \Big|$$
$$\bullet P \quad \Big| \quad \mathsf{0} \quad \Big| \quad P + P \quad \Big| \quad !\, P \quad \Big| \quad \partial P$$

Les trois premières constructions permettent de "transporter" à l'ordre supérieur les opérateurs déjà définis sur les structures d'actions. Quant aux cinq dernières, elles représentent les termes de contrôle.

Le constructeur $\bullet$, appelé *comittal*, a pour fonction de dénoter la continuation d'un processus (après une action). On l'utilise pour modéliser une transition : supposons que la réaction d'un processus P donne des résultats de la forme $a. \bullet P'$ (on peut trouver à la section suivante la définition de la dynamique des processus). L'action a pouvant se décomposer sous la forme $a^*.e$, e effet, la réaction peut s'écrire : $P \searrow a^*.e. \bullet P'$. Réaction qui peut s'interpréter, si $P'' \equiv e.P'$, comme la transition de P à P'' par l'action a^* :

$$P \xrightarrow{a^*} P''$$

On dit que e est un effet de l'action a, et P'' est le résultat de l'effet sur P'. Milner propose d'ailleurs dans [13], une structure d'action pour CCS dans laquelle $\bullet$ correspond exactement à la composition d'une garde à un processus $(\alpha \cdot P)$. Le constructeur ∂ quant à lui, est hérité de CCS synchrone (SCCS). Il peut être interprété comme un opérateur de "délai" [12]. C'est-à-dire qu'il est utilisé, dans ∂P, pour retarder les réductions sur P[6]. Il sert par exemple à définir l'opérateur de cmposition parallèle :

$$P \parallel Q \stackrel{def}{=} P \otimes \partial Q + \partial P \otimes Q$$

$\mathsf{0}$ (nul), $+$ (somme) et ! (réplication) ne nécessitent aucune explication.

2.4 Dynamique des structures de processus.

L'approche modulaire qui a été choisie: on a ajouté à la définition d'une structure d'actions un ensemble de constructeurs munis de leur propre dynamique, permet d'éclairer la sémantique des calculs de processus. En particulier elle rend naturelle la classification des réductions en deux types. Conjoncturelle : deux acteurs d'un même processus interagissent; c'est la réaction. Ou structurelle : la réduction est le résultat d'un changement dans la structure des termes; c'est la réduction d'une structure de contrôle.

Un processus P peut être interprété comme une action d'arité $n \to \epsilon$. Et le processus $a.P$ $(a : m \to n)$, comme le processus qui "débute" par l'action a,

[6] l'interprétation des opérateurs de contrôle sera éclaircie par la donnée de leur dynamique dans la section suivante

les effets de cette action étant transmis à P. Si $n \neq \epsilon$, P est paramétré par les effets produits par a, "l'information" pouvant être transmise continuement au cours des multiples réactions. La composition ainsi définie est très proche de la composition sur les structures d'actions, à la différence qu'on ne permet pas de composer deux processus par ".".

La réaction définie sur la structure d'action se prolonge à la structure de processus. Il nous faut cependant définir la notion de contexte *ouvert*[7] : L'occurrence d'une action a, ou d'un processus Q, dans P est dite *ouverte* si elle n'est pas à l'intérieur d'un terme de contrôle. Par exemple, P apparaît "ouvert" dans $P \otimes Q$, mais pas dans $P + Q$. Un contexte $\mathcal{C}[\,]$ est alors dit ouvert, si pour tout P, P apparaît ouvert dans $\mathcal{C}[P]$. On peut voir un contexte ouvert comme un contexte qui permet une réduction "conjoncturelle". On définit également une congruence structurelle sur l'ensemble des processus, $\equiv$, qui est la plus petite congruence vérifiant:

$$\begin{array}{rcl} P & \equiv & id.P \\ a.(b.P) & \equiv & (a.b).P \\ (a.P) \otimes (b.Q) & \equiv & (a \otimes b).(P \otimes Q) \\ (x)(a.P) & \equiv & (ab_x a).(x)P \end{array}$$

Définition (dynamique des actions).
$\searrow_a$ est le plus petit préordre, clos par congruence structurelle, tel que:

$$a \searrow_A a' \text{ et } \mathcal{C}[\,] \text{ contexte ouvert} \qquad \Rightarrow \qquad \mathcal{C}[a.P] \searrow_a \mathcal{C}[a'.P]$$

la réduction des structures de contrôle est elle aussi définie par la donnée d'un préordre: $\searrow_c$.

Définition (dynamique des structures de contrôle).
$\searrow_c$ est le plus petit préordre, clos par congruence structurelle, tel que:

$$\begin{array}{cc} \bullet P \otimes \bullet Q \searrow_c \bullet(P \otimes Q) & (x) \bullet P \searrow_c \bullet(x)P \\ P + Q \searrow_c P & P + Q \searrow_c Q \\ \partial P \searrow_c P & \partial P \searrow_c \bullet\partial P \\ !P \searrow_c P \otimes !P & !P \searrow_c \bullet !P \end{array}$$

De plus si $P \searrow_c P'$ et $\mathcal{C}[\,]$ est un contexte ouvert,

$$\mathcal{C}[P] \searrow_c \mathcal{C}[P']$$

Comme annoncé en introduction, la réunion de ces deux réductions nous donne la relation que l'on rencontre dans les calculs de processus, avec une légère différence dûe au traitement intégré des effets dans les structures d'actions.

[7] dans [13] et [14], Milner parle de "ready context".

Définition (Relation de transition).

On définit à partir de toute structure d'action A, la relation de transition $\{\xrightarrow{a} / a \in A\}$ sur $\mathcal{P}_A$ comme étant la plus petite relation satisfaisant:

$$\frac{P \searrow a^*.e.\bullet P'}{P \xrightarrow{a^*} e.P'}$$

où $\searrow = (\searrow_a \cup \searrow_c)^*$.

`Exemple :` On peut étudier le cas de la communication (pour rendre l'exemple plus parlant on utilisera l'action silencieuse τ) :

$$\bar{y}x.P \parallel y(z).Q \xrightarrow{\tau} P \parallel Q\{x/z\}$$

Avec notre modélisation, cet exemple se réécrit

$$\begin{array}{l} ((\underline{\pi}.\bullet P) \otimes \partial(\pi.\bullet Q)) \quad + \\ \qquad (\partial(\underline{\pi}.\bullet P) \otimes (\pi.\bullet Q)) \end{array} \quad \xrightarrow{\tau} \quad P \otimes Q\{x/z\}$$

où $\underline{\pi}$ est une action qui signifie envoyer x sur le canal y, et π signifie recevoir un nom sur y et le lier à z (π peut être vu comme la composition de deux actions). Cette transition est dérivée de l'ensemble de réductions :

$$\begin{array}{lcl} ((\underline{\pi}.\bullet P) \otimes \partial(\pi.\bullet Q)) \quad + & & \\ \qquad (\partial(\underline{\pi}.\bullet P) \otimes (\pi.\bullet Q)) & \searrow_c & \underline{\pi}.\bullet P \otimes \partial\pi.\bullet Q \\ & \searrow_c & \underline{\pi}.\bullet P \otimes \pi.\bullet Q \\ & \equiv & (\underline{\pi} \otimes \pi).(\bullet P \otimes \bullet Q) \\ & \searrow_c & (\underline{\pi} \otimes \pi).\bullet(P \otimes Q) \\ & \searrow_a & \tau.(id \otimes \{x/z\}).\bullet(P \otimes Q) \end{array}$$

et $(id \otimes \{x/z\}).(P \otimes Q) \equiv (P \otimes Q\{x/z\})$

Cette relation de transition, qui permet de définir de manière uniforme sur les calculs de processus une relation de bisimulation, possède des propriétés intéressantes. Propriétés que nous tenterons de retrouver indépendamment et avec une approche différente dans la suite de notre étude. En particulier on a les propositions suivantes :

$$\begin{array}{ll} \text{si } P \xrightarrow{a} P' & \\ & c.P \xrightarrow{c.a} P' \\ & \text{si } a \searrow a', \qquad P \xrightarrow{a'} P' \\ & \text{si } a = a'.e, \qquad P \xrightarrow{a'} e.P' \\ & \text{si } Q \xrightarrow{b} Q', \qquad P \otimes Q \xrightarrow{a \otimes b} P' \otimes Q' \end{array}$$

Ces propriétés soulignent l'intérêt des structures d'actions. En effet la définition d'une structure de processus se fait de manière uniforme, elles permettent donc de prouver des propriétés générale.

2.5 PIC: une structure d'action pour le π-calcul.

Nous allons nous intéresser à PIC, une structure d'actions pour le π-calcul définie de manière très détaillée dans [13][8]. Les constituants élémentaires de PIC, appelés *particules*, formalisent les “événements” observables dans le π-calcul: l'émission et la réception d'un nom sur un canal, ainsi que la restriction de la portée d'un nom.

$$\text{(particules)} \qquad \pi ::= \quad x(\overrightarrow{y}) \quad \big| \quad \bar{x}\langle\overrightarrow{y}\rangle \quad \big| \quad \nu\ x$$

Une action a de PIC d'arité $m \to n$ est de la forme :

$$a \quad = \quad (\overrightarrow{x})S\langle\overrightarrow{y}\rangle$$

où S est une séquence $[\pi_1 \dots \pi_k]$ de particules, $\overrightarrow{x}$ est un vecteur de m noms distincts et $\overrightarrow{y}$ est un vecteur de n noms. La particule $x(y)$ lie le nom y dans chaque séquence qui la contient[9], la portée de ce lien s'étendant “à droite” de son point d'occurrence. Ainsi la séquence $[\dots x(y)\bar{y}\langle z\rangle \dots]$ modélise la réception d'un nom y, suivi de son utilisation comme canal d'émission. Une particule π qui ne contient pas y peut donc être considérée comme indépendante de $x(y)$. On note cette relation $x(y) \smile \pi$. Cette relation s'étend à une liste de particules en posant que $[\dots x(y)\pi \dots] \smile [\dots \pi x(y) \dots]$. Une séquence représente donc la classe modulo $\smile$ d'une liste de particules. De même les noms importés (les noms de $\overrightarrow{x}$) sont liés dans S, et cet effet se prolonge aux noms exportés (noms de $\overrightarrow{y}$). On considérera de plus deux actions égales à α-conversion près comme équivalentes.

Dans cette brève introduction, nous avons défini la forme des actions, ainsi que les ensembles d'actions $A_{m,n}$ (voir section 2.1). Pour terminer de définir PIC de manière exacte, il nous faut encore donner :

– L'ensemble de noms X. Il est égal à l'ensemble des noms du π-calcul.

– Le monoïde $(M, \otimes, \epsilon)$ des arités, qui est l'ensemble des entiers naturels munis de l'addition.

– L'ensemble des actions neutres : $id_n = (x_1 \dots x_n)[]\langle x_1 \dots x_n\rangle$

[8] On étudie ici le π-calcul polyadique, dans lequel on communique, plutôt qu'un nom unique, un vecteur

[9] le même phénomène existe avec le nom x dans $\nu\ x$

- Les différents opérateurs. Soient $a = (\vec{u})S\langle\vec{v}\rangle$ et $b = (\vec{x})T\langle\vec{y}\rangle$, telles qu'aucun nom lié ne soit partagé, on désigne par $\frown$ la concaténation des séquences (modulo $\smile$).

$$\begin{array}{rcl} a.b & \stackrel{def}{=} & (\vec{u})(S \frown T\{\vec{v}/\vec{x}\})\langle\vec{y}\{\vec{v}/\vec{x}\}\rangle \\ a \otimes b & \stackrel{def}{=} & (\vec{u}\,\vec{x})S \frown T\langle\vec{v}\,\vec{y}\rangle \\ ab_x a & \stackrel{def}{=} & (x\,\vec{u})S\langle x\,\vec{v}\rangle \end{array}$$

- La réaction. Soit $a = (\vec{u})S\langle\vec{v}\rangle$, avec $S \smile (S_1 \frown [\bar{x}\langle\vec{y}\rangle x(\vec{z})] \frown S_2)$. $\searrow$ est définie par les axiomes de base et la donnée de la réaction élémentaire :

$$a \quad \searrow \quad (\vec{u})S_1 \frown S_2\{\vec{y}/\vec{z}\}\langle\vec{v}\{\vec{y}/\vec{z}\}\rangle$$

  Cette réaction capture la notion de communication du π-calcul. En particulier

$$(\ )[x(y)]\langle y\rangle \otimes (z)[\bar{x}z]\langle\ \rangle \quad \searrow \quad (z)[\emptyset]\langle z\rangle = id_1$$

  qu'il faut comparer à

$$x(y).P \parallel \bar{x}z.Q \longrightarrow P\{z/y\} \parallel Q$$

On peut d'ailleurs simplifier la définition de PIC en introduisant un ensemble d'actions "atomiques". Ces actions (définies ci-dessous) ont une interprétation intuitive frappante, intuition que nous retrouverons en partie avec l'équivalence par permutations. Leurs intérêt est de permettre d'engendrer toutes les actions de PIC, tout en leur donnant une interprétation sous forme de composition d'actes élémentaires.

émission	$out_x : m \to 0$	$\stackrel{def}{=}$	$(\vec{y})[\bar{x}\langle\vec{y}\rangle]\langle\ \rangle$	$(\lvert\vec{y}\rvert = m)$
réception	$in_x : 0 \to m$	$\stackrel{def}{=}$	$(\ )[x(\vec{y})]\langle\vec{y}\rangle$	$(\lvert\vec{y}\rvert = m)$
valeur x	$\langle x\rangle : 0 \to 1$	$\stackrel{def}{=}$	$(\ )[\ ]\langle x\rangle$	
nom privé	$\nu : 0 \to 1$	$\stackrel{def}{=}$	$(\ )[\nu\ x]\langle x\rangle$	
destruction	$\omega : 1 \to 0$	$\stackrel{def}{=}$	$(x)[\ ]\langle\ \rangle$	

On peut cependant remarquer que, bien que PIC soit considéré comme une structure d'action pour le π-calcul par Milner, il ne partage que peu de points communs avec le π-calcul présenté en introduction (figure 1.1). En particulier la notion de garde de communication a disparu en faveur du concept "d'*enablement*" : une communication peut se produire dès que les variables (les noms liés) sont suffisamment instanciées. Pour retrouver la sémantique du π-calcul, une idée est d'utiliser les résultats de la section 2.3. Il s'agit de construire la structure de processus associée a PIC : $\mathcal{P}_{\mathrm{PIC}}$, en n'oubliant pas d'ajouter aux définitions déja données celles des constructeurs particuliers au π-calcul (par exemple le $\|$ et la restriction). Mais, comme nous l'avions déja remarqué, l'algèbre ainsi obtenue n'est plus une structure d'actions.[10]

$$\begin{array}{lcl}
P \parallel Q & \stackrel{def}{=} & (P \otimes \partial Q) + (\partial P \otimes Q) \\
\bar{x}\,\vec{y}.P & \stackrel{def}{=} & \langle \vec{y} \rangle.out_x.\bullet \partial P \\
x(\vec{y}).P & \stackrel{def}{=} & in_x.(\vec{y}) \bullet \partial P \\
(\nu x)P & \stackrel{def}{=} & \nu.(x)P \\
\\
\langle \vec{y} \rangle & \stackrel{def}{=} & \langle y_1 \rangle \otimes \ldots \otimes \langle y_m \rangle \\
(\vec{y})(\;) & \stackrel{def}{=} & (y_1)\ldots(y_m)(\;)
\end{array}$$

[10] À titre d'idée, le lecteur pourra reprendre l'exemple de la communication en notant que :

$$\begin{array}{lcll}
\bar{x}y.P & = & (\;)[\;]\langle y \rangle \;.\; (u)[\bar{x}\langle u \rangle]\langle\;\rangle \;.\bullet \partial P & \\
x(z).Q & = & (\;)[x(z)]\langle z \rangle \;.\; (z) \bullet \partial Q & (\searrow_c (\;)[x(z)]\langle z \rangle \;.\; \bullet (z)Q)
\end{array}$$

Chapitre 3

Équivalence par permutation.

Nous allons définir dans cette partie, pour les besoins de notre étude, deux sémantiques opérationnelles pour le π-calcul. Ce type de sémantique a rencontré un grand succès dans la modélisation de langages tels que CCS. On spécifie, par un ensemble de règles structurelles, un système de transitions étiquetées. Un calcul peut alors être considéré comme l'enchaînement de pas de réduction élémentaires d'un état à un autre : $e \xrightarrow{\alpha} e'$, l'étiquette de cette transition mémorisant le "type" d'action qui a été effectué. Suivant l'idée de Boudol et Castellani [7][6], nous allons étudier un système de transitions dans lequel les étiquettes seront les preuves des transitions de la S.O.S. de départ. La description d'un processus devient alors le système des séquences de *transitions prouvées* qu'il engendre, modulo une relation d'équivalence entre séquences qui permet d'échanger l'occurrence de deux *actions indépendantes*. C'est l'équivalence par permutations, telle que l'ont définie Berry et Levy pour le λ-calcul[2].

Remarque : Les sémantiques opérationnelles décrites dans ce chapitre sont toutes réunies en annexe afin de faciliter leur lecture et leur étude comparée.

3.1 Système de transitions prouvées pour le π-calcul.

3.1.1 Transitions avec préinstantiation (EPIC).

La sémantique avec préinstantiation EPIC (pour "Early instantiation semantic" pour le PI Calcul)[1] définie quatre types d'actions dans le π-calcul.

[1] Cette sémantique est équivalente a celle proposée par Milner & al dans [18]

$$\begin{array}{rl} \theta ::= & x(y) \;\;\mathbf{|}\;\; \bar{x}y \;\;\mathbf{|}\;\; \theta \mid id \;\;\mathbf{|} \\ & \quad id \mid \theta \;\;\mathbf{|}\;\; \theta \mid \theta \;\;\mathbf{|}\;\; (x)\theta \;\;\mathbf{|} \\ & \qquad \Box\theta \;\;\mathbf{|}\;\; (\bigtriangledown x)\theta \;\;\mathbf{|}\;\; x \triangleright \theta \end{array}$$

TAB. 3.1 - syntaxe des preuves

Dont l'émission $\bar{x}w$, la réception $x(y)$ et la communication τ. La règle la plus intéressante, `Open`, concerne l'émission de nom restreint. Elle définit, avec la règle `Close`, la notion d'émission de nom privé $\bar{x}(y)$. Ces règles permettent en particulier de représenter le partage d'un nom privé, comme la règle suivante le permet dans la sémantique standard du π-calcul.

$$((\nu\ x)\ P) \parallel Q \equiv (\nu\ x)\ (P \parallel Q) \qquad (x \notin \mathrm{fn}(Q))$$

Remarque : par souci de concision nous n'avons pas défini l'ensemble des règles de nos sémantiques. En effet la plupart des règles possèdent des symétriques qui n'ont pas été données. De plus la sémantique du "bang" est légèrement différente de celle du π-calcul, puisque la réplication est dirigée par les réactions. *i.e.* on n'a pas de règle du type : $\vdash\ !P \longrightarrow P \parallel !P$.

3.1.2 Transitions prouvées pour EPIC.

Le système EPIC nous fournit une étiquette pour chaque transition entre deux termes du π-calcul, mais il ne fournit pas le schéma d'inférence utilisé pour étiqueter la transition. *i.e.* , il ne nous dit pas quelles sont les règles que nous avons utilisées pour inférer la transition, et comment nous les avons employées. Dans le système de transitions prouvées, on garde la preuve de la transition comme étiquette. Ceci nous permet de conserver un maximum d'information, en particulier on verra qu'on mémorise le site, dans l'arbre syntaxique d'un terme, d'où "provient" l'action. Ce résultat découle en partie du choix fait dans notre sémantique de ne pas considérer de congruence structurelle sur les termes[2].

Plutôt que de mémoriser le nom des règles employées, nous allons utiliser un ensemble de symbôles qui définissent la syntaxe des preuves. Celle-ci est donnée par la grammaire de la table 3.1 (il faut remarquer que certaines preuves obtenues par cette grammaire ne correspondent à aucune transition de EPIC).

- $x(y)$, $\bar{x}y$ dénotent les actions élémentaires.

[2] Dans le π-calcul standard, $\equiv$ fait de $(\mathcal{P}, \parallel)$ un monoïde commutatif. Dans ce système $P \parallel Q \xrightarrow{\alpha} P' \parallel Q$ et $Q \parallel P \xrightarrow{\alpha} Q \parallel P'$ sont donc les mêmes transitions.

– $\theta \ | \ id$ dénote l'utilisation de la règle sur le parallèle, on définit également son symétrique : $id \ | \ \theta$.

– $(x)\theta$ mémorise l'utilisation de la règle sur la restriction.

– $\square\theta$ et $x \triangleright \theta$ mémorisent l'emploi de la règle `Open`. $\square$ permet de signaler la levée d'une restriction, tandis que $\triangleright$ signale que l'on propage sa portée (règles `Propagate`).

– $(\triangledown x)\theta$ dénote l'utilisation de la règle `Close`.

On définit également pour les besoins de notre sémantique, les notions de nom de canal (cl) et de valeur (vl). On montre en particulier que pour toute transition prouvée étiquetée par θ, $cl(\theta)$ et $vl(\theta)$ permettent de retrouver l'étiquette de la transition associée de EPIC.

θ		$cl(\theta)$	$vl(\theta)$
$x(w)$		x	w
$\bar{x}w$		$\bar{x}$	w
$\theta \ \| \ id$	$id \ \| \ \theta$	$cl(\theta)$	$vl(\theta)$
$(x)\ \theta$	$\square\theta$	$cl(\theta)$	$vl(\theta)$
$x \triangleright \theta$		$cl(\theta)$	x
$\theta_1 \ \| \ \theta_2$	$(\triangledown x)\theta$	τ	τ

`Exemple :` Soit le terme :

$$P \overset{def}{=} (\nu x)\ (x(y).R_1 \parallel \bar{x}z.R_2)$$

P peut se réduire suite à une communication au port x selon la transition :

$$P \xrightarrow{\tau} (\nu x)\ (R_1\{z/y\} \parallel R_2)$$

qui se déduit de :

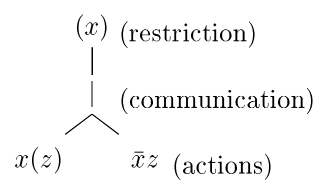
$$\frac{\dfrac{\dfrac{}{x(y).R_1 \xrightarrow{x(z)} R_1\{z/y\}\ \text{(Action)}} \qquad \dfrac{}{\bar{x}z.R_2 \xrightarrow{\bar{x}z} R_2\ \text{(Action)}}}{x(y).R_1 \parallel \bar{x}z.R_2 \xrightarrow{\tau} R_1\{z/y\} \parallel R_2\ \text{(Communication)}}}{P \xrightarrow{\tau} (\nu x)\ (R_1\{z/y\} \parallel R_2)\ \text{(Restriction)}}$$

soit, en utilisant les notations de la table 3.1 :

$$P \xrightarrow{(x)(x(z) \;\mid\; \bar{x}z)} (\nu x)\ (R_1\{z/y\} \parallel R_2)$$

On voit sur cet exemple que la sémantique avec transitions prouvées nous fournit le résultat escompté. *i.e.* elle donne des étiquettes dont l'arbre syntaxique reflète exactement l'arbre de preuve de la transition :

$(x)(x(z) \;\mid\; \bar{x}z) \simeq$



On peut également noter que la preuve d'une transition nous fournit toute l'information associée à la dérivation. Ceci au sens que la donnée du processus de départ P, et de la preuve θ, suffit à déterminer de manière unique la transition (*i.e.* P', tel que $P \xrightarrow{\theta} P'$, est uniquement fonction de P et θ).

Avec l'aide de nos nouvelles notations nous allons être à même de souligner une faiblesse d'EPIC. Comme nous l'avons vu en introduction de cette section, nous avons muni la sémantique opérationnelle des deux règles `Open` et `Close` afin de permettre l'extension de la portée des restrictions. Mais la sémantique ne précise pas que toute utilisation de la règle `Open` doit être suivie de l'emploi de `Close`. Ainsi il est possible de dériver la transition suivante dans EPIC[3] :

$$(\star) \qquad (\nu z)(\bar{x}z.0 \parallel \bar{y}z.0) \xrightarrow{u \,\triangleright\, (\bar{x}u \;\mid\; id)} 0 \parallel \bar{y}u.0$$

Ce résultat étonnant, puisqu'il nous dit qu'il est possible de faire disparaître une restriction, ne modifie pourtant pas la validité de nos choix. En effet cette transition revient à déplacer la restriction "en tête" de processus. Ce qui est équivalent à :

$$\ldots \parallel (\nu x)P \parallel \ldots \quad \equiv \quad (\nu y)(\ldots \parallel P\{y/x\} \parallel \ldots) \qquad y \text{ nouveau nom}$$

[3] On peut cependant noter que le système de transitions prouvées ne permet pas d'inférer des transitions du type :

$$(\nu z)(\bar{x}z.0 \parallel \bar{y}z.0) \parallel Q \xrightarrow{u \,\triangleright\, (\bar{x}u \;\mid\; id) \;\mid\; id} (0q \parallel \bar{y}u.0) \parallel Q$$

Il va cependant nous falloir rester prudent car la relation de transition n'est plus compositionnelle. En particulier on n'a pas l'implication :

$$P \xrightarrow{\theta} P' \quad \Rightarrow \quad P \parallel Q \xrightarrow{\theta \ | \ id} P' \parallel Q$$

Pour s'en persuader il suffit de considérer la transition ($\star$).

3.1.3 Transitions avec postinstantiation (LPIC).

La définition d'une sémantique opérationnelle différente d'EPIC pour le π-calcul est motivée par le souci de se rapprocher des structures d'actions. En effet dans la sémantique avec préinstantiation il faut considérer une infinité de réductions du type $x(y).P \xrightarrow{x(w)} P\{w/y\}$, alors qu' au sens intuitif il s'agit essentiellement de la même action. On introduit dans cette partie deux nouveaux constructeurs. L'abstraction: "$(y).P$", et la concrétion : "$[w].P$". L'abstraction se rapproche de ce qui est défini dans le λ-calcul, elle permet de lier un nom dans P et de le transformer en variable. La concrétion est son pendant, dans le sens qu'elle permet d'instancier les variables d'une abstraction. On peut trouver dans [15] des exemples d'utilisation de ces deux nouvelles formes qui montrent tout leur intérêt. Avec ces deux nouvelles constructions, la communication est remplacée par ce que Milner appelle un *commitment*, et qui est symbolisée par un nouveau signe, $\bullet$. On peut voir dans la règle de `Commitment` l'équivalent de la β-réduction.

Mais cette sémantique pose différents problèmes. Tout d'abord elle permet de distribuer la composition parallèle au cours d'une rencontre (loi `Distribute`). La position relative des termes n'est donc plus préservée. Ceci pose un problème pour la définition du système de transitions prouvées, car les preuves ne permettent plus alors de mémoriser l'emplacement des réactions. Un autre problème soulevé par la définition de LPIC, est l'absence de règles permettant de dériver sous les abstractions et les concrétions. Cette absence semble a-priori justifiée, puisque la sémantique standard du π-calcul nous interdit de réduire un terme sous une garde. Cependant la séquence des deux transitions $\bar{x}.\bar{x}.P \xrightarrow{\bar{x}} \bar{x}.P \xrightarrow{\bar{x}} P$, par exemple, est autorisée dans EPIC. Et, comme nous le verrons en introduisant la notion de séquences de transition, cette possibilité fait tout l'intérêt de notre étude.

3.2 Permutation de transitions concurrentes

Nous sommes maintenant munis d'un système de transitions pour le π-calcul[4], dans lequel nous pouvons étudier les différentes transitions, ainsi que leurs séquences, observables à partir d'un processus fixé P :

[4]On utilisera EPIC dans la suite de notre étude.

$$\begin{array}{llll} P \xrightarrow{\theta_{1,0}} P_{1,0} & P_{1,0} \xrightarrow{\theta_{1,1}} P_{1,1} & P_{1,1} \xrightarrow{\theta_{1,2}} P_{1,2} & \ldots \\ P \xrightarrow{\theta_{2,0}} P_{2,0} & P_{2,0} \xrightarrow{\theta_{2,1}} P_{2,1} & \ldots & \\ \ldots & & & \end{array}$$

Cet ensemble est nommé $\mathcal{T}(P)$:

`Définition (Séquence de transitions).`

Pour tout terme P du π-calcul, on définit $\mathcal{T}(P)$ comme étant l'ensemble des séquences de la forme:

$$P \xrightarrow{\theta_1} P_1 \xrightarrow{\theta_2} \ldots \xrightarrow{\theta_{n-1}} P_{n-1} \xrightarrow{\theta_n} P_n$$

C'est l'ensemble des "calculs" finis $t_1 \ldots t_n$ ($P_0 = P$ et $t_i = P_{i-1} \xrightarrow{\theta_i} P_i$) observables à partir de P. On peut y définir l'opération de concaténation qui à tout s_1 (resp. s_2) élément de $\mathcal{T}(P)$ (resp. $\mathcal{T}(Q)$), tel que s_1 se termine en Q, associe $s_1.s_2 \in \mathcal{T}(P)$.

Intuitivement, on aimerait pouvoir définir deux transitions comme indépendantes, ou compatibles, quand leurs réalisations ne s'excluent pas réciproquement. Par exemple deux communications ne seraient pas compatibles du moment qu'elles partagent un même "participant". Cette intuition peut se formaliser grâce aux notations introduites dans la section précédente. Avant de définir la relation de *concurrence* entre deux transitions prouvées, c'est le terme que nous choisirons pour la suite, on définira tout d'abord la notion de concurrence entre preuves. En particulier deux transitions seront *concurrentes* si les preuves qui les étiquettent le sont.

`Définition (preuves concurrentes).`

On définit la relation $\smile$ comme étant la plus petite relation symétrique vérifiant les axiomes $\mathcal{C}_1$ à $\mathcal{C}_4$[5].

$$(\mathcal{C}_1) \qquad \forall \theta \qquad id \smile \theta$$

$$(\mathcal{C}_2) \qquad \begin{array}{l} \forall \theta_1 \smile {\theta_1}' \\ \forall \theta_2 \smile {\theta_2}' \end{array} \qquad \theta_1 \mid \theta_2 \smile {\theta_1}' \mid {\theta_2}'$$

$$(\mathcal{C}_3) \qquad \forall \theta \smile \theta' \qquad (x)\ \theta \smile (x)\ \theta'$$

[5] On considèrera que id est une preuve. $\mathcal{C}_1$ implique donc que $id \smile id$

$$(\mathcal{C}_4) \qquad \forall\theta \smile \theta' \qquad \frac{x \triangleright \theta \smile \theta'}{(\triangledown x)\ \theta \smile \theta'}$$

Définition (Transitions concurrentes).
Soient $t_0 = P \xrightarrow{\theta_0} P_0$ et $t_1 = P \xrightarrow{\theta_1} P_1$ deux transitions prouvées. t_0 et t_1 sont des transitions concurrentes, on note $t_0 \smile t_1$, si et seulement si $\theta_0 \smile \theta_1$.

On peut montrer, conformément à notre intuition, que deux transitions concurrentes peuvent être observées dans "n'importe quel ordre". *i.e.* qu'elles représentent deux actions qui peuvent être appliquées successivement à un processus (indépendamment de l'ordre d'occurrence) sans modifier le résultat (voir schéma 3.1). L'idée est que deux transitions à partir d'un même processus, $P \xrightarrow{\theta_1} P_1$ et $P \xrightarrow{\theta_2} P_2$ avec $\theta_1 \smile \theta_2$, peuvent être mises en séquence. Mais dans ce cas, et contrairement à la théorie des traces développée par Mazurkiewicz [10], le résiduel d'une action n'est pas l'action elle même. *i.e.* on n'obtient pas la séquence $P \xrightarrow{\theta_1} P' \xrightarrow{\theta_2} P''$, mais une séquence $P \xrightarrow{\theta_1} P' \xrightarrow{\theta_2{}'} P''$, où $\theta_2{}'$ est ce qu'il reste de la transition θ_2 après que θ_1 se soit réalisée. On parle de résidu d'une transition.

`Exemple` : Soit le terme P défini par :

$$P \stackrel{def}{=} ((\nu x)\bar{y}x.R_1 \parallel \bar{z}.R_2) \parallel y(u).R_3$$

On peut définir les deux transitions concurrentes θ_1 et θ_2, qui représentent la première une communication et la seconde une émission.

$$\begin{array}{lcl} \theta_1 & \stackrel{def}{=} & (\triangledown x)((\square\bar{y}x \mid id) \mid y(x)) \\ \theta_2 & \stackrel{def}{=} & (id \mid \bar{z}) \mid id \end{array}$$

On peut également définir la séquence de ces deux transitions :

$$P \xrightarrow{\theta_1} (\nu x)((R_1 \parallel \bar{z}.R_2) \parallel R_3\{x/u\}) \xrightarrow{\theta_2{}'} (\nu x)((R_1 \parallel R_2) \parallel R_3\{x/u\})$$

$$\theta_2{}' = (x)((id \mid \bar{z}) \mid id) \neq \theta_2$$

L'exemple précédent nous montre comment deux actions concurrentes peuvent "s'affecter" (ici c'est par le mouvement d'une restriction). Une étude plus précise des règles qui définissent le résidu d'une preuve permet de prouver que le déplacement des restrictions est le seul responsable de ce phénomène. En particulier, pour des transitions sans "mouvement de restrictions", résidus et preuves sont égaux[6]. Mais avant de définir le résiduel de deux preuves, on peut remarquer qu'avec notre définition de la concurrence il est impossible d'inférer les implications :

$$\theta \smile \theta' \quad \Rightarrow \quad \begin{array}{ll} \square\theta \smile \square\theta' & (\star) \\ \square\theta \smile (x)\theta' & (\star\star) \end{array}$$

On peut déduire deux conséquences de cette remarque:

1° Deux transitions qui modifient la même restriction ne peuvent pas être concurrentes (d'après $(\star)$).

2° Une transition qui modifie la portée de la restriction d'un nom à un sous-processus, ne peut pas être concurrente à une transition dont l'effet touche ce sous-processus (d'après $(\star\star)$).

Ces propriétés de la relation de concurrence, et en particulier la seconde, semblent contredire l'intuition que nous avions donné au début de ce chapitre. Mais il n'est pas possible d'inclure $(\star)$ et $(\star\star)$ comme règles dans la définition de $\smile$ sans créer d'exemple incohérent. Nous reviendrons sur ce point après quelques définitions.

La notion de résiduel peut être formalisée en utilisant les preuves, de la même manière que pour la notion de concurrence.

Définition (Résiduel).

Le résiduel d'une preuve θ' par une preuve θ ($\theta' \smile \theta$) est noté θ'/θ. Il est définit de manière inductive par l'ensemble des règles $(\mathcal{R}_1)$ à $(\mathcal{R}_6)$

$(\mathcal{R}_1)$ $\quad id/\theta = id$

$(\mathcal{R}_2)$ $\quad \theta'/id = \theta'$

$(\mathcal{R}_3)$ $\quad (\theta_1{}' \mid \theta_2{}')/(\theta_1 \mid \theta_2) = (\theta_1{}'/\theta_1) \mid (\theta_2{}'/\theta_2)$

$(\mathcal{R}_4)$ $\quad (x)\theta'/(x)\theta = (x)(\theta'/\theta)$

$(\mathcal{R}_5)$ $\quad (\bigtriangledown x)\theta'/\theta = (\bigtriangledown x)(\theta'/\theta)$
$\quad \theta'/(\bigtriangledown x)\theta = (x)(\theta'/\theta)$

$(\mathcal{R}_6)$ $\quad x \triangleright \theta'/\theta = x \triangleright (\theta'/\theta)$
$\quad \theta'/x \triangleright \theta = \theta'/\theta$

[6] *i.e.* $\theta/\theta' = \theta$ avec les notations de la définition suivante

On peut, à partir de ces définitions, donner un résultat sur la confluence "partielle" de notre système de transitions. En effet on peut démontrer que dans le cas de deux transitions concurrentes, on peut trouver des transitions qui, mises en séquence, permettent de retrouver le même résultat. Et plus encore. On sait donner ces transitions puisqu'il s'agit des résidus.

`Lemme ( Confluence locale ).`

Soient $t_0 = P \xrightarrow{\theta} P_\theta$ et $t_1 = P \xrightarrow{\theta'} P_{\theta'}$ deux transitions prouvées concurrentes. Alors il existe un unique terme du π-calcul, P' tel que: $P_\theta \xrightarrow{\theta'/\theta} P'$ et $P_{\theta'} \xrightarrow{\theta/\theta'} P'$.

Une interprétation graphique (figure 3.1) souligne utilement le résultat énoncé par ce lemme.

Cette propriété nous permet de définir une équivalence sur $\mathcal{T}(P)$ qui égale les séquences dans lesquelles on permute des transitions concurrentes. Elle capture l'intuition que, lors d'une exécution complexe, on peut commuter deux pas de réduction indépendants. On retrouve ici l'idée de Mazurkiewicz qui, le premier, a défini une sémantique à partir d'une équivalence sur les séquences de transitions[7].

`Définition (Équivalence par permutations).`

On appelle équivalence par permutations, $\equiv$, la plus petite relation d'équivalence sur $\mathcal{T}(P)$ qui vérifie:

$$
\begin{array}{lllll}
(1) & t \smile t' & \Rightarrow & t.(t'/t) \equiv t'.(t/t') \\
(2) & s_1 \equiv {s_1}' & & \\
 & s_2 \equiv {s_2}' & & \\
 & s_1.s_2 \text{ définie} & \Rightarrow & s_1.s_2 \equiv {s_1}'.s_2 \\
 & & & s_1.s_2 \equiv s_1.{s_2}'
\end{array}
$$

On peut remarquer que cette équivalence conserve des propriétés intéressantes sur les séquences de transitions. En particulier la taille, ainsi que le "terme d'arrivée" d'une séquence sont laissés invariants. De plus, comme dans la théorie des traces, on peut représenter un élément de $\mathcal{T}(P)$ par un ordre partiel. En effet, si on considère qu'une séquence de transitions ordonne ses composants (suivant l'ordre naturel induit par la concaténation), la classe d'équivalence d'une séquence peut être représentée par l'intersection des ordres totaux induis par chacune des séquences équivalentes.

`Exemple :`

Considérons le terme P défini par

[7] Sémantique pour les réseaux de Pétri [10]

$$
\begin{tikzcd}
 & P_\theta \arrow[dr, "\theta'/\theta"] & \\
P \arrow[ur, "\theta"] \arrow[dr, "\theta'"'] & & P' \\
 & P_{\theta'} \arrow[ur, "\theta/\theta'"'] &
\end{tikzcd}
$$

FIG. 3.1 - propriété de Church-Rosser pour les séquences de transitions.

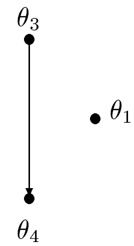
$$(\nu\ z)(\ \bar{x}z.0 \parallel z(v).0\ )\ \parallel x(u).\bar{u}t.0$$

On peut donner quelques exemples de transitions dérivables à partir de P et de ses produits.

$$
\begin{array}{rcl}
\theta_1 & = & z \rhd (\square(\bar{x}z \ \mid\ id) \ \mid\ id) \\
\theta_2 & = & (\bigtriangledown w)(\square(\bar{x}w \ \mid\ id) \ \mid\ x(w)) \\
\theta_3 & = & id \ \mid\ x(w) \\
\theta_4 & = & id \ \mid\ \bar{w}t
\end{array}
$$

Ceux-ci illustrent quelques aspects du système que nous avons développé jusqu'ici. θ_1 et θ_2 fournissent un exemple de deux transitions non concurrentes, tandis que $\theta_1 \smile \theta_3$. Enfin on a l'exemple d'une séquence de transitions non concurrentes, car causalement reliées : $\theta_3.\theta_4$.

Un élément de $\mathcal{T}(P)/\equiv$ peut se représenter sous la forme du graphe d'un ordre partiel dont les sommets sont les transitions, et dont la relation est "calquée" sur la relation de dépendance entre transitions. Ainsi on peut représenter la classe d'équivalence de la séquence $\theta_1.(\theta_3/\theta_1).\theta_4$ par le graphe :



Ce graphe indique que les seules séquences équivalentes à $\theta_1.(\theta_3/\theta_1).\theta_4$ dans $\mathcal{T}(P)$ sont $\theta_3.(\theta_1/\theta_3).\theta_4$ et $\theta_3.\theta_4.(\theta_1/\theta_3)$.

◇

3.3 Réduction dans $\mathcal{T}(P)$.

Le but de cette section est de définir une relation sur $\mathcal{T}(P)/\equiv$ qui traduise, à l'échelle "macroscopique", la notion de réduction sur les termes du π-calcul. En particulier la communication. De façon intuitive, il s'agit de définir une relation sur les séquences de transitions telle que, toute séquence dans laquelle une émission et une réception se produisent de manière concurrente, via le même nom, puisse se réduire en une séquence qui comporte une communication.

3.3.1 Réduction de deux transitions concurrentes.

La première étape consiste à étudier la notion de réduction sur une séquence de deux transitions concurrentes. Ou plus simplement l'étude de la réduction de deux transitions. Afin d'améliorer la compréhension des mécanismes de réduction dans le π-calcul, il est pratique de faire apparaître une classification des transitions. Dans les différentes sémantiques proposées jusqu'à maintenant, chaque transition pouvait être intuitivement interprétée comme appartenant à l'une des cinq catégories suivantes :

$(in_{x(y)})$	Réception sur un canal
$(out_{\bar{x}w})$	Émission sur un canal
(τ_x)	Communication
(νout_x)	Émission d'un nom privé sur un canal
$(\nu\tau_x)$	Communication d'un nom privé

Chacune de ces catégories peut être définie plus abstraitement comme une sous-partie de l'ensemble des transitions prouvées engendrée par les grammaires suivantes (on se réfère ici plus précisément à la sémantique EPIC). Par exemple :

$$
\begin{array}{lll}
in_{x(y)} ::= & x(y) \;\Big|\; in_{x(y)} \mid id \;\Big| \\
 & \qquad id \mid in_{x(y)} \;\Big|\; \forall z \notin \{x, y\},\ (z) in_{x(y)} \\[2ex]
out_{\bar{x}w} ::= & \bar{x}w \;\Big|\; out_{\bar{x}w} \mid id \;\Big| \\
 & \qquad id \mid out_{\bar{x}w} \;\Big|\; \forall z \notin \{x, w\},\ (z) out_{\bar{x}w} \\[2ex]
\tau_x ::= & out_{\bar{x}y} \mid in_{x(y)} \;\Big|\; in_{x(y)} \mid out_{\bar{x}y} \;\Big| \\
 & \qquad \tau_x \mid id \;\Big|\; id \mid \tau_x \;\Big|\; \forall z,\ (z)\tau_x
\end{array}
$$

Et en posant: $\left\{ \begin{array}{ll} \sigma_{x,y} ::= & \Box out_{\bar{x}y} \;\Big|\; \sigma_{x,y} \mid id \;\Big| \\ & id \mid \sigma_{x,y} \;\Big|\; \forall z \notin \{x, y\},\ (z)\sigma_{x,y} \end{array} \right.$

$$
\begin{array}{lll}
\nu out_x ::= & y \triangleright \sigma_{x,y} \\[2ex]
\nu\tau_x ::= & (\bigtriangledown y)(\sigma_{x,y} \mid in_{x(y)}) \;\Big|\; (\bigtriangledown y)(in_{x(y)} \mid \sigma_{x,y}) \;\Big| \\
 & \qquad \nu\tau_x \mid id \;\Big|\; id \mid \nu\tau_x \;\Big|\; \forall z,\ (z)\tau_x
\end{array}
$$

Ces classes forment une partition de l'ensemble des transitions prouvées. Dans le sens que, si Θ est l'ensemble des transitions observables à partir de termes quelconques du π-calcul:

$$\Theta = \bigcup_{x,y} in_{x(y)} \oplus \bigcup_{x,y} out_{\bar{x}y} \oplus \bigcup_{x} \tau_x \oplus \bigcup_{x} \nu out_x \oplus \bigcup_{x} \nu\tau_x$$

Cette décomposition donne exactement les transitions que l'on peut observer le long d'un calcul. Ce qui nous donne un ensemble plus “fin” que celui obtenu par la grammaire des preuves (voir table 3.1).

L'explicitation mathématique de l'intuition donnée précédemment, est facilitée par les définitions des différentes classes de transitions. En effet notre but est de définir une relation (notons la $\searrow$) à même de vérifier des implications du type:

$$\theta \in out_{\bar{x}y},\ \theta' \in in_{x(y)},\ \theta \smile \theta' \qquad \Rightarrow \qquad \theta.(\theta'/\theta) \searrow \theta_0 \text{ et } \theta_0 \in \tau_x$$

ce qui correspond à l'égalité $out_x \otimes in_x \searrow \tau$ de la structure d'action pour le π-calcul.

On définit la fonction partielle Φ qui fournit le résultat de la réduction de deux transitions concurrentes. Φ est défini récursivement par l'ensemble des règles suivantes.

$$\Phi : \Theta^2 \to \Theta \qquad \left\{ \begin{array}{l} \Phi((x)\theta, (x)\theta') = (x)\Phi(\theta, \theta') \\ \Phi(\theta \mid id, \theta' \mid id) = (\Phi(\theta, \theta')) \mid id \\ \Phi(x \triangleright \theta, \theta') = (\bigtriangledown x)\Phi(\theta, \theta') \\ \\ \Phi(\theta \mid id, id \mid \theta') = \theta \mid \theta' \\ \qquad\qquad\qquad \text{si } \exists x, (\theta \mid \theta') \in \tau_x \end{array} \right.$$

On démontre que Φ est à valeur dans Θ, i.e qu'elle réduit deux transitions prouvées en une transition qui correspond à la réduction d'un terme du π-calcul. En fait elle réduit une réception/émission en une communication et une réception/émission de nom privé en une communication d'un nom privé. Dans tout autre cas, Φ n'est pas définie.

3.3.2 Réduction de séquences de transitions prouvées, et problèmes.

A partir de Φ il est facile de définir la réduction d'une séquence $\theta.(\theta'/\theta)$ de deux transitions concurrentes. Il suffit de la remplacer par la séquence à une seule transition: $\Phi(\theta, \theta')$ $(= \Phi(\theta', \theta))$.

Un bon candidat pour la relation de réduction recherchée semble donc être $\searrow$, obtenu en prolongeant Φ à l'ensemble des séquences de transitions prouvées (et en restant compatible avec $\equiv$). *i.e.* , $\searrow$ est le plus petit préordre qui vérifie:

$$\begin{array}{llcl} (1) & \theta \smile \theta' & & \\ & \Phi(\theta,\theta') \text{ défini} & \vdash & \theta.(\theta'/\theta) \searrow \Phi(\theta,\theta') \\ & & & \theta'.(\theta/\theta') \searrow \Phi(\theta,\theta') \\ (2) & \forall s', \forall s \searrow s_0 & & s'.s \searrow s'.s_0 \\ & & & s.s' \searrow s_0.s' \\ (3) & s \equiv s_0 \searrow s_0' \equiv s' & \vdash & s \searrow s' \end{array}$$

Mais cette relation n'est pas stable dans $\mathcal{T}(P)$, comme nous le montre l'exemple suivant.

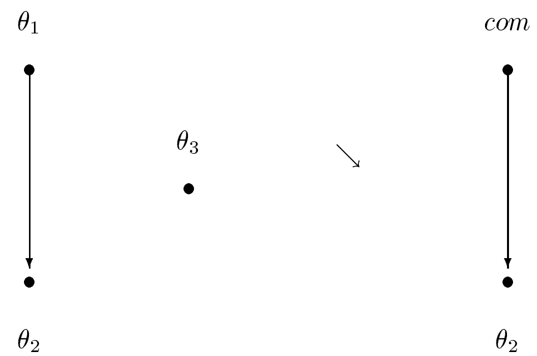
`Exemple :`
Étudions l'exemple de la création d'un lien privé entre deux processus

$$P \ = \ (\nu\ z)(\ \bar{x}z.z(u).R_1\ ) \parallel x(y).\bar{y}v.R_2$$

soit

$$\begin{array}{lcll} \theta_1 & = & z \triangleright (\square \bar{x}z) \ \mid \ id) & \in \nu out_x \\ \theta_2 & = & z(u) \ \mid \ id & \in in_{z(u)} \\ \theta_3 & = & id \ \mid \ x(z) & \in in_{x(z)} \end{array}$$

la séquence $\theta_1.\theta_2.\theta_3$ de $\mathcal{T}(P)$ peut se réduire en effectuant la communication au port x ($\searrow$ possède une interprétation graphique très parlante sous forme de réduction de graphes) :



où $\quad com = (\bigtriangledown z)(\square \bar{x}z \ \mid \ x(y)) \quad (\in \nu\tau_x) \quad$. Mais, alors que la transition étiquettée par *com* est observable à partir de P, La séquence $[com.\theta_2]$ n'est pas dans $\mathcal{T}(P)$. En effet la communication a modifiée la restriction, et c'est la séquence $com.[z \triangleright \square(z(u) \ \mid \ id)]$ qu'on devrait observer.

--- ◇

Les problèmes que nous rencontrons avec la réduction sur $\mathcal{T}(P)$ sont liés à la définition "locale" que nous en avons faite. En effet il aurait été étonnant que la communication, qui agit "syntaxiquement" sur tout P (on emploie la "métaopération" de substitution sur l'ensemble des termes) n'ait, au niveau des transitions, qu'un effet local. On peut énumérer ces problèmes.

1° Après une communication, il n'y a pas substitution des noms au niveau des transitions. D'ailleurs, les transitions ne permettent pas de traduire la notion de substitution comme "effet" d'une communication. Problème qu'il sera important de résoudre lorsque nous voudrons étudier les rapports avec les structures d'actions. Une conséquence de ce problème, est qu'il est possible d'obtenir, après réduction, une séquence de transitions qui n'appartient plus à $\mathcal{T}(P)$. Ainsi dans l'exemple précédent :

$$\theta_1.\theta_2.\theta_3.[(id \ \mid \ \bar{y}v)] \quad \searrow \quad com.\theta_2.[(id \ \mid \ \bar{y}v)]$$

alors que la séquence de $\mathcal{T}(P)$ correspondante devrait être :

$$com.[z \triangleright (\Box(z(u) \ \mid \ id))].[(id \ \mid \ \bar{z}v)]$$

2° Lors de l'émission d'un nom privé, on enlève une restriction sans, plus tard, la remettre. Comme on le fait dans une communication avec la règle `Close`. Ceci pose un grave problème, puisqu'une réduction peut "refermer" la restriction, l'effet de cette réduction ne modifiant pas le reste des transitions de la séquence. Cet effet est mis en évidence par le dernier exemple présenté. En effet :

$$\theta_1.\theta_2.\theta_3 \quad \searrow \quad [(\nabla z)(\Box\bar{x}z \ \mid \ x(y))].[z(u) \ \mid \ id]$$

alors qu'on s'attend à obtenir $[(\nabla z)(\Box\bar{x}z \ \mid \ x(y))].[z \triangleright (\Box(z(u) \ \mid \ id))]$.

3° On autorise les communications sous les gardes (voir exemple précédent), contrairement à la sémantique standard du π-calcul.

Dans le chapitre suivant, nous tenterons de trouver les solutions pouvant être apportés à ces problèmes. Ceci en ayant pour but de définir une structur d'actions pour le π-calcul à partir de l'équivalence par permutations. On peut cependant tenter de résumer ces solutions.

- Se munir d'une notion de substitution au niveau des séquences de transitions.
- "Coder" dans notre sémantique opérationnelle les effets du π-calcul. *i.e.* la substitution de noms, la levée des gardes et le mouvement des restrictions.

Mais la réduction n'est pas notre seul problème. Revenons à notre remarque sur les choix fait pour $\smile$. Sans les limitations que nous nous sommes imposées, le lemme formulant la "confluence locale" de notre système de transitions n'est plus vrai. Et donc l'équivalence par permutation ne peut plus être définie. En effet en ajoutant les axiomes $(\star)$ et $(\star\star)$ [8] à la définition de la relation de concurrence, les phénomènes de substitution liés aux restrictions entraînent des incohérence dans la définition des résidus. Si on étudie, par exemple, le processus

$$P \stackrel{def}{=} (\nu z)(\bar{x}z \parallel \bar{y}z)$$

---

[8] $$\theta \smile \theta' \quad \Rightarrow \quad \begin{array}{ll} \Box\theta \smile \Box\theta' & (\star) \\ \Box\theta \smile (x)\theta' & (\star\star) \end{array}$$

On peut observer les transitions étiquetées par :

$$\begin{array}{rcl} \theta_1 & = & u \triangleright \square(\bar{x}u \mid id) \\ \theta_2 & = & v \triangleright \square(id \mid \bar{y}v) \end{array}$$

$$\begin{tikzcd} & P \arrow[dl, "\theta_1"'] \arrow[dr, "\theta_2"] & \\ 0 \parallel \bar{y}u & & \bar{x}v \parallel 0 \end{tikzcd}$$

On voit bien qu'après θ_1 et θ_2, il n'est pas possible de se ramener en une seule transition au même processus (alors que $(\star)$ implique $\theta_1 \smile \theta_2$). En effet le nom z lié dans P est remplacé par un nom qui est libre. C'est cette même raison qui impose de ne pas choisir $(\star\star)$ comme axiome[9]. Il nous faudra donc aussi régler le problème des transitions qui "lèvent" une restriction. En particulier, plutôt que de s'interdire les transitions du type $x \triangleright \theta$, on créera des transitions dont le seul but est de déplacer les restrictions.

[9]Le lien est plus subtil : Il y a conflit avec l'axiome $\mathcal{R}_6$ pour définir le résidu. En effet on ne peut pas se contenter de poser : $(\square\theta')/(x)\theta = \square(\theta'/\theta)$ et $(x)\theta'/(\square\theta) = (\theta'/\theta)$

Chapitre 4

Étude comparée des deux modèles.

Nous allons nous intéresser à une étude comparée des structures d'actions et de l'équivalence par permutations pour le π-calcul. Pour ce faire, nous allons légèrement modifier la structure des preuves que nous avons introduite au chapitre précédent.

4.1 Étude de FREEPIC

4.1.1 Développement sur la restriction : la classe de transitions ν

Nos choix dans les différentes sémantiques choisies pour le π-calcul, ainsi que dans la définition de l'équivalence par permutations, ne sont pas uniques. Motivé par l'étude des structures d'actions, et plus particulièrement par la définition de Milner dans [16] d'une "action calculi" pour le π-calcul[1], nous allons présenter une sémantique, et l'équivalence par permutations associée, qui met l'accent sur le rôle de la restriction. Ce choix est également motivé par la difficulté de modélisation du comportement de la restriction, difficulté que nous avons pu observer dans les chapitres précedents.

On ajoute à la sémantique avec préinstantiation les règles `ScopeMigration` et `ScopeClosure`:

[1] Le "calcul d'actions" pour le π-calcul, est une algèbre particulière possédant, en plus des opérateurs des structures d'actions, trois *contrôles*, dont ν qui représente la création de nom nouveau.

ScopeClosure:
$$P \xrightarrow{x \,\triangleright\, \theta} P' \qquad \vdash \qquad P \xrightarrow{(\bigtriangledown x)\ \theta} (\nu\ x)\ P'$$

ScopeMigration:
$$(\nu x)P \xrightarrow{y \,\triangleright\, \square} P\{y/x\} \quad y \notin \mathrm{fn}(P)$$

Ces règles nous permettent de définir une nouvelle classe de transitions prouvées pour notre calcul. Cette classe, que nous appellerons "ν", représente les transitions correspondantes à la modification du domaine d'une restriction. *i.e.* les transitions qui lèvent une restriction et la referment. Comme précedemment on peut donner leur grammaire.

En posant:
$$\nu_x ::= \quad \square \quad \Big| \quad \nu_x \mid id \quad \Big| \quad id \mid \nu_x \quad \Big| \quad \forall z \neq x, (z)\nu_x$$
on obtient:
$$\nu ::= \quad (\bigtriangledown x)\nu_x \quad \Big| \quad \nu \mid id \quad \Big| \quad id \mid \nu \quad \Big| \quad \forall z,\ (z)\nu$$

L'introduction de ces règles nous offre deux possibilités. La première consiste à étudier LPIC augmentée des règles sur la restriction. Cette solution a le désavantage de rendre le système de règles redondant. En effet les règles `Open`, `Propagate` et `Close` se superposent aux nouvelles. Ou retirer de la sémantique les règles devenues redondantes. C'est ce second choix que nous allons développer dans le reste de notre étude.

4.1.2 Transitions dans FREEPIC

FREEPIC est la sémantique dérivée de LPIC en remplaçant les règles `Open`, `Propagate` et `Close` par les règles `ScopeMigration` et `ScopeClosure`. Comme pour LPIC, on peut définir la notion de concurrence et de résiduel[2]. Avec une telle sémantique, un échange de nom privé n'est plus modélisable par une unique transition, mais par une séquence de deux transitions prouvées non concurrentes: la première pour modifier le champ de la restriction, suivie d'une transition pour la communication. Ce choix entraîne l'apparition d'une dépendance "temporelle" à l'intérieur des séquences de transitions; c'était déjà le cas dans LPIC lors d'une réaction "sous une garde" (voir exemple section 3.2).

On ajoute également deux nouvelles constructions à la grammaire des transitions prouvées (cf. annexe A). Ceci conformément à ce qui avait été annoncé

[2]On confondra par la suite FREEPIC avec la sémantique des "transitions prouvées" qu'elle induit

au chapitre précédent.

- "$\bullet$" pour représenter la levée d'une garde par une communication. On pourrait intuitivement dire que

$$[x]P \xrightarrow{\bullet} P$$

- "$\{x/y\}$" pour la substitution et la levée d'une garde par une réception :

$$(y)P \xrightarrow{\{x/y\}} P\{x/y\}$$

La syntaxe de l'ensemble des preuves dénotant une communication se retrouve donc légèrement modifiée. Les notations ayant volontairement été choisies les plus parlantes possibles. Nous allons d'ailleurs définir un nouveau "type" de preuves de transitions, les *substitutions*, afin de remplir un des buts que nous nous étions fixés au chapitre précédent. Les substitutions étant étroitement liées aux communications, nous avons choisi d'utiliser la même syntaxe (l'emploi du constructeur $\langle\ \rangle$ les distinguant).

Définition (substitution).
On associe à chaque preuve θ de τ la substitution $\langle\theta\rangle$, et on nomme $\langle\tau\rangle$ leur ensemble. Une substitution peut être appliquée à une preuve pour donner une nouvelle preuve[3] *:*

$$\begin{array}{lcl}
\theta\langle\{x/y\}\rangle & = & \theta\{x/y\}^3 \\
(\theta \mid \theta')\langle s \mid s'\rangle & = & \theta\langle s\rangle \mid \theta'\langle s'\rangle \\
\theta\langle\bullet\rangle \ = \ \theta\langle id\rangle & = & \theta \\
 & & \\
((x)\theta)\langle(x)s\rangle & = & (x)(\theta\langle s\rangle) \\
 & & \\
(\Box\theta)\langle s\rangle & = & \Box\theta^3 \\
(x \triangleright \theta)\langle s\rangle & = & x \triangleright \theta \\
((\nabla x)\theta)\langle s\rangle & = & (\nabla x)\theta
\end{array}$$

4.2 Actions = transitions prouvées.

Nous avons jusqu'à maintenant étudié les preuves comme étant les étiquettes de transitions observables à partir d'un terme du π-calcul. De même les séquences de transitions prouvées étaient définies comme les séquences de calcul d'un terme P, modulo l'équivalence par permutations ($\mathcal{T}(P)/\equiv$). Or il est possible de considérer l' "algèbre" des preuves, que nous appellerons désormais

[3] On ne substitue pas de nom lié. On remarque également que les preuves de ν ne sont pas affectées par la substitution.

actions pour souligner l'interprétation différente que nous voulons en faire, indépendamment de l'étude des calculs d'un processus fixé. Il nous faut cependant définir avec plus de précision ce qu'est une action, et les opérations qui agissent sur elles.

Définition (actions).

On définit l'ensemble des actions, $\mathcal{A}ct$, comme étant le plus petit ensemble contenant l'ensemble des preuves des transitions de FREEPIC : Θ et clos par l'opération de composition séquentielle $\odot$ [4]*. En particulier :*

$$\Theta = \nu \oplus \bigcup_{x,y} in_{x(y)} \oplus \bigcup_{x,y} out_{\bar{x}y} \oplus \bigcup_{x} \tau_x \qquad \subset \qquad \mathcal{A}ct$$

Remarque Les preuves représentant l'émission d'un nom privé ne sont pas des actions (on n'accepte pas de preuves du type $x \triangleright \theta$). On s'épargne donc les problèmes rencontrés au chapitre précédent avec la restriction.

L'introduction des actions correspond à notre volonté de pouvoir considérer une preuve comme un programme qui, partant d'un processus, fournirait un processus résultat. Nous avons défini comme actions, et nous avons noté $\mathcal{A}ct$ leur ensemble, aussi bien les preuves élémentaires que les "séquences" de preuves. Cependant notre opération de composition séquentielle sur les actions n'est pas toujours définie. En effet deux actions a et b ne peuvent être composées que s'il existe un processus P tel que a (respect. b) peut être interprété comme une séquence de transitions de P à P' (respect. P' à P''). Leur composition $a \odot b$, correspondant alors à une séquence de transitions de P à P''. C'est pour formaliser cette condition de compatibilité entre actions, qui peut se comparer aux conditions posées sur les arités dans les structures d'actions, que nous introduisons la notion d'interprétation.

Définition (interprétation d'une action).

On définit pour chaque action a de $\mathcal{A}ct$ son interprétation, $\mathcal{I}(a)$, comme étant l'ensemble des couples de processus (P, P') tels que "$P \xrightarrow{a} P'$", i.e. *:*

$$\begin{array}{lcl} \theta \textit{ preuve} \in \mathcal{A}ct, \quad \mathcal{I}(\theta) & = & \{(P,P')/P \xrightarrow{\theta} P'\} \\ a = b \odot c, \qquad \mathcal{I}(a) & = & \{(P,P'')/\exists P', (P,P') \in \mathcal{I}(b) \textit{ et } (P',P'') \in \mathcal{I}(c)\} \end{array}$$

$(\mathcal{A}ct, \odot)$, dans lequel la composition n'est autorisée qu'entre actions d'interprétation convenable, est alors isomorphe à l'ensemble des séquences de transitions prouvées muni de la concaténation.

[4]cette opération, qui reflète la concaténation sur les séquences de transition aux niveau des preuves, sera définie avec plus de précision par la suite

Il est également possible de définir une opération de composition parallèle sur $\mathcal{A}ct$, qui à tout couple d'actions (a, b) associe une nouvelle action $a \otimes b$. La définition de l'interprétation, prolongée aux nouvelles actions, permet d'éclairer cette opération :

Soit $a, b \in \mathcal{A}ct$, $\mathcal{I}(a \otimes b) = \{ (P \parallel Q, P' \parallel Q') \quad / (P, P') \in \mathcal{I}(a), (Q, Q') \in \mathcal{I}(b)\}$

On peut d'ailleurs interpréter $\otimes$ très simplement en utilisant les constructeurs définis jusqu'à présent. En effet si on étend la définition du constructeur $|$ au niveau des séquences de preuves en posant que[5] :

$$\forall \theta \text{ preuve de } \Theta \quad [\theta] \mid id = [\theta \mid id] \qquad (\theta \mid id) \in \Theta$$

$$\forall a, b \in \mathcal{A}ct \qquad (a \odot b) \mid id = (a \mid id) \odot (b \mid id)$$

On obtient :

$$a \otimes b \stackrel{def}{=} (a \mid id) \odot (id \mid b)$$

Pour retrouver la définition classique d'une structure algébrique, il nous faut également introduire l'élément neutre : $\mathbb{1}_{\mathcal{A}ct}$. On caractérisera $\mathbb{1}_{\mathcal{A}ct}$ par la séquence de preuves de taille nulle, [ ]. En effet aucune preuve de Θ ne peut le représenter, car aucune ne possède d'interprétation cohérente[6]. On prend comme axiome les propriétés qui font de $\mathbb{1}_{\mathcal{A}ct}$ l'élément neutre de $\odot$ et $\otimes$:

$$\forall a \in \mathcal{A}ct \qquad \mathbb{1}_{\mathcal{A}ct} \odot a = a \odot \mathbb{1}_{\mathcal{A}ct} = a \qquad \mathbb{1}_{\mathcal{A}ct} \otimes a = a \otimes \mathbb{1}_{\mathcal{A}ct} = a$$

L'égalité employée dans la formule précédente peut être interprétée comme une relation d'équivalence ($\sim$) sur $(\mathcal{A}ct, \otimes, \odot)$. Relation que l'on peut raffiner afin de se rapprocher des propriétés des structures d'actions. En particulier on considérera par la suite que :

- les séquences $[(\theta_1 \mid \theta_2) \mid \theta_3]$ et $[\theta_1 \mid (\theta_2 \mid \theta_3)]$ sont égales.
- deux séquences "congruentes" modulo équivalence par permutations sont égales. En particulier

$$\begin{array}{lll} \theta \smile \theta' & \Rightarrow & [\theta] \odot [\theta'/\theta] \sim [\theta'] \odot [\theta/\theta'] \\ a \sim a' & \Rightarrow & a \odot b \sim a' \odot b \\ & & \ldots \\ & & b \otimes a \sim b \otimes a' \end{array}$$

[5] on utilise les crochets ("[","]") pour dénoter les séquences composées d'une preuve, et on considère id comme un élément de $\mathcal{A}ct$

[6] on devrait par exemple avoir $\mathcal{I}(a \otimes id) = \mathcal{I}(a)$

On peut d'ailleurs remarquer que, en employant une égalité moins fine que l'égalité syntaxique sur les termes (on doit considérer $P \parallel (Q \parallel R)$ et $(P \parallel Q) \parallel R$ comme étant deux termes égaux), deux actions équivalentes selon $\sim$ ont même interprétation. La définition des actions, que nous avions donné en introduction de ce chapitre se retrouve éclairée par ces nouvelles notions.

`Définition (l'ensemble` $(\mathcal{A}ct, \otimes, \odot)$`).`
$\mathcal{A}ct$ est défini comme le plus petit ensemble contenant Θ, ensemble des actions élémentaires, augmenté de l'action id et de l'élément neutre ($\mathbb{1}_{\mathcal{A}ct}$), et stable par les opérations de composition séquentielle ($\odot$) et parallèle ($\otimes$). De plus on confondra deux actions équivalentes selon $\sim$.

4.2.1 Rapport avec les structures d'action.

On peut démontrer à partir de nos définitions quelques propriétés proches des axiomes des structures d'actions. Par exemple:

- existence d'un élément neutre pour $\otimes$.
- Associativité. Pour $\odot$ tout d'abord, puisque la concaténation sur les séquences de transitions est associative par définition :

$$\forall a,b,c \in \mathcal{A}ct, \quad \begin{array}{l}(a \odot b)\\ (b \odot c) \text{ défini}\end{array} \quad \Rightarrow \quad (a \odot b) \odot c = a \odot (b \odot c)$$

  Et pour $\otimes$ puisque :

$$(a \otimes b) \otimes c \sim ((a \mid id \mid id) \odot (id \mid b \mid id)) \odot (id \mid id \mid c)$$

- Distributivité

$$(a \odot b), (c \odot d) \text{ défini} \quad \Rightarrow \quad \begin{array}{l}(a \odot b) \otimes (c \odot d) = (a \otimes c) \odot (b \otimes d)\\ \text{et } (a \otimes c) \odot (b \otimes d) \text{ défini}\end{array}$$

Mais la définition d'une notion d'arité sur les actions pose problème. On peut par exemple choisir l'ensemble des entiers naturels comme monoïde des arités, et considérer les actions définies jusqu'à présent comme étant d'arité $0 \rightarrow 0$. Il suffit alors de définir les notions de noms exportés et importés (ou abstraits) afin de se ramener à un traitement comparable à celui vu pour la structure d'actions PIC. Ce choix à l'avantage d'être simple et de permettre la définition d'un opérateur très intuitif pour l'abstraction. Mais il n'intègre pas la condition qui restreint la composition séquentielle. *i.e.* deux actions peuvent être composées si et seulement si leurs interprétations sont d'intersection non nulle.

La définition de l'arité que nous allons choisir par la suite tient compte de ce défaut. On définit les actions comme des triplets $P \xrightarrow{a} P'$ avec $a \in \mathcal{A}ct$, et $(P, P') \in \mathcal{I}(a)$. L'arité de $P \xrightarrow{a} P'$ est alors $P \to P'$. Le monoïde des arités devient $(\mathcal{P}, \|, \epsilon)$, où $\mathcal{P}$ est le plus petit ensemble stable par l'opérateur parallèle, contenant ϵ ainsi que tous les termes du π-calcul.

Remarque ϵ, l'élément neutre de ce monoïde, ne peut pas être choisi parmi les termes du π-calcul. En effet il doit vérifier la propriété $P \| \epsilon = P$. De plus $\mathbb{1}_{\mathcal{A}ct}$ correspondra à l'action $\epsilon \xrightarrow{\mathbb{1}_{\mathcal{A}ct}} \epsilon$. On remarque également que l'on peut étendre notre congruence structurelle, $\sim$, au nouveau type d'action que nous considérons[7]. Mais, afin de faciliter nos définitions ultérieures, on ajoute deux axiomes à sa définition qui permettent de "raffiner" l'écriture des actions, soit :

$$\begin{array}{lll} (P, P') \in \mathcal{I}(a) & & \\ (P', P'') \in \mathcal{I}(b) & \Rightarrow & P \xrightarrow{a \ \odot \ b} P'' \sim (P \xrightarrow{a} P') \ \odot \ (P' \xrightarrow{b} P'') \end{array}$$

$$\begin{array}{lll} (P, P') \in \mathcal{I}(a) & & \\ (Q, Q') \in \mathcal{I}(b) & \Rightarrow & P \| Q \xrightarrow{a \otimes b} P' \| Q' \sim (P \xrightarrow{a} P') \otimes (Q \xrightarrow{b} Q') \end{array}$$

Cette notation "a du sens". En effet à processus P et preuve θ fixés, on sait que le processus résultat de la transition (*i.e.* P' tel que $P \xrightarrow{\theta} P'$) est défini de manière unique. Ce résultat, qui s'étend naturellement aux séquences de preuves, nous démontre donc que l'on ne peut pas raffiner les actions aléatoirement.

4.2.2 Relation de réduction sur les actions.

La définition de la relation de concurrence ($\smile$) sur les séquences de $\mathcal{T}(P)$ se traduit aisément dans ($\mathcal{A}ct$, $\odot$). Mais la traduction n'est pas aussi facile pour la relation de réduction. En effet il nous faut pouvoir faire interagir deux actions situées de part et d'autre d'un "$\|$", et permettre la substitution des noms au niveau des actions. C'est le rôle des éléments de $\langle\tau\rangle$, introduit au début de ce chapitre. Lors d'une réduction, on remplace une émission et une réception concurrentes par une communication. La preuve de cette communication code la substitution que nous allons propager (les notations introduites pour FREEPIC sont parlantes). A cette fin on définit l'effet d'une substitution sur une action à partir de son effet sur une preuve :

$$\begin{array}{ll} \forall\theta \text{ preuve de } \Theta & [\theta]\langle s\rangle = [\theta\langle s\rangle] \\ \\ \forall a, b \in \mathcal{A}ct & (a \ \odot \ b)\langle s\rangle = (a\langle s\rangle) \ \odot \ (b\langle s\rangle) \\ & (a \otimes b)\langle s \ \mid \ s'\rangle = (a\langle s\rangle) \otimes (b\langle s'\rangle) \end{array}$$

[7]Ce qui, par exemple, a pour conséquence de transformer $\|$ en opérateur associatif

Mais il faut rester attentif. En effet il ne suffit pas d'appliquer la substitution aux éléments de la séquence d'actions apparaissant "à droite" de la communication : il nous faut tenir compte de la possibilité de permutation entre actions concurrentes. Afin de faire face à cette nouvelle difficulté, nous allons définir la réduction de deux preuves concurrentes par la donnée du couple : communication résultat, substitution induite[8]. Comme avec les séquences de transitions prouvées du chapitre précédent, nous allons utiliser une fonction auxiliaire Φ pour définir la réduction :

$$\Phi((x)\theta, (x)\theta') = (x)\Phi(\theta, \theta')$$
$$\Phi(\theta \mid id, \theta' \mid id) = (\Phi(\theta, \theta')) \mid id$$

$$\Phi(\theta \mid id, id \mid \theta') = [\theta \mid \theta'] \odot [\langle \theta \mid \theta' \rangle] \quad \text{si } \exists x, (\theta \mid \theta') \in \tau$$

Définition (Réduction sur les actions).
$\searrow$ est le plus petit préordre sur l'ensemble des actions, compatible avec $\sim$ et qui vérifie :

$$(1) \quad \Phi(\theta, \theta') \textit{ existe} \quad \vdash \quad (P \xrightarrow{[\theta]} P') \odot (P' \xrightarrow{[\theta']} P'') \searrow P \xrightarrow{\Phi(\theta,\,\theta')} P''$$

$$(2) \quad \forall \theta, \theta' \quad \Phi(\theta, \theta') \textit{ existe} \quad \vdash \quad (P \xrightarrow{[\theta]} P') \otimes (Q \xrightarrow{[\theta']} Q') \searrow P \parallel Q \xrightarrow{\Phi(\theta \mid id,\, id \mid \theta')} P' \parallel Q'$$

$$(3) \quad \forall S \searrow S_0, \forall S' \qquad \begin{array}{lcl} S' \odot S & \searrow & S' \odot S_0 \\ S' \otimes S & \searrow & S' \otimes S_0 \\ S \odot S' & \searrow & S_0 \odot S' \\ S \otimes S' & \searrow & S_0 \otimes S' \end{array}$$

et afin de tenir compte de l'effet des substitutions :

$$(4) \quad P \xrightarrow{[\langle s \rangle]} P' \odot P' \xrightarrow{[\theta]} P'' \searrow P \xrightarrow{[\langle s \rangle] \odot [\theta \langle s \rangle] \odot [\langle s \rangle]} P''$$

Remarque la duplication de la substitution dans la règle (4) est nécessaire. En effet il faut garder la trace de la réaction afin d'agir sur des actions qui, par permutations, seraient déplacées après le point d'occurrence de la communication. On peut cependant considérer que $[\langle s \rangle] \odot [\langle s \rangle] \sim [\langle s \rangle]$.

[8]On peut voir la substitution comme une continuation qui transforme les séquences d'actions pour leur appliquer l'effet d'une communication.

Nous définirons, dans la suite de notre étude, les substitutions comme des actions à part entière. C'est d'ailleurs ce que nous avions déjà implicitement considéré dans les définitions précédentes. Ainsi pour toute action $P \xrightarrow{[\theta]} P'$ dénotant une communication (*i.e.* $\theta \in \tau$), on définit l'action $P \xrightarrow{[\langle\theta\rangle]} P'$, de même arité. En étendant la définition de la concurrence a ces nouvelles actions, on obtient quelques propositions intéressantes qui valident nos choix sur la communication. Ainsi, si on considère que :

$$\theta \smile \theta' \quad \Rightarrow \quad \langle\theta\rangle \smile \theta'$$

on peut démontrer qu'une substitution ne peut pas entrer en conflit avec une autre : deux substitutions concurrentes peuvent être permutées sans modifier leurs effets respectifs. De même une substitution et la communication qui l'"engendre" ne peuvent pas être concurrentes. Ceci remplit notre premier objectif, qui était d'appliquer la substitution aux seules actions pouvant se dérouler après la communication. On remarque également que la substitution d'un nom dans une action demande plusieurs pas de réduction, alors que dans la sémantique standard du π-calcul il s'agissait d'une métaopération.

On retrouve ici encore des propriétés proches de celle de la réaction sur les structures d'actions:

- La réaction préserve l'arité d'une action.
- Les opérateurs préservent la réaction.
- En considérant que $P \xrightarrow{id_P} P$ est l'action neutre d'arité $P \to P$. Avec :

$$\begin{array}{lcll} id_\epsilon & = & \mathbb{1}_{\mathcal{A}ct} & \\ id_{P_1 \| P_2} & = & id_{P_1} \mid id_{P_2} & \\ id_{(\nu x)P} & = & (x)id_P & \\ id_\pi & = & id & (\pi \text{ particule}) \end{array}$$

  On obtient de manière évidente :

$$[id_P] \searrow S \qquad \Longleftrightarrow \qquad S = [id_P]$$

Mais de nombreux problèmes subsistent, qui ne nous permettent pas de prétendre avoir défini une structure d'action :

1° il nous manque une famille d'opérateurs d'abstraction. Ce problème n'est pas simple, ou tout du moins ne possède pas de solution naturelle.

2° le choix de $(\mathcal{P}, \|, \epsilon)$ pour définir l'arité des actions n'est pas très "intuitif". En effet la signification intuitive de l'arité dans les structures d'actions est celle d'un système de typage statique (ou *sort*), dans lequel chaque nom possède un type[9]

3° les structures d'actions se veulent une algèbre unifiant la syntaxe avec la sémantique. En utilisant la syntaxe du π-calcul dans la définition de nos actions on réalise bien ce résultat. Mais au prix d'une lourdeur dans la définition des actions.

[9] Dans PIC on utilise les entiers naturels pour mémoriser le nombre de valeurs importées et exportées. Un nom ayant le sort 1

Chapitre 5

Conclusion

C'est sur un bilan mitigé que se finit l'étude comparée de nos deux modèles. En effet, si nous avons réussi a souligner les liens entre structures d'actions et équivalence par permutations, nous n'avons pas réussit à définir une structure d'actions simple et “naturelle” pour le π-calcul. Nous aurons néanmoins, je l'espère, réussi à rendre plus intuitive la définition des structures d'actions. Par exemple, le sens de la composition séquentielle et du paramétrage des actions sont éclaircis par l'opérateur de concaténation entre séquences de transitions.

Notre étude aura également souligné l'importance de la restriction dans la sémantique du π-calcul. Alors que dans la sémantique standard, l'utilisation d'une congruence structurelle sur les termes simplifie la communication, avec EPIC et LPIC la gestion des restrictions devient explicite : on intègre la gestion des restrictions à la sémantique opérationnelle. Nous aurons aussi, d'une manière plus subtile, permis de cerner le statut central que possèdent les noms dans le π-calcul. En effet les complications que nous avons rencontrées peuvent également s'expliquer par le rôle trop important donné à la syntaxe des termes dans l'équivalence par permutations. On peut d'ailleurs trouver de nombreuses tentatives pour donner une sémantique du π-calcul dénué de ce problème[1].

Ce travail aura révélé un problème riche en sujet d'étude. Nous l'avons vu pour la restrictions et la notion de causalité[2] dans le π-calcul. Mais de nombreuses recherches restent aussi à faire. En particulier pour définir des opérateurs d'abstractions pour notre modèle, et pour redéfinir l'arité, dont le choix

[1] Dans [13], on peut trouver une modélisation “graphique” du π-calcul “dépourvue” d'opérateur de restriction. Dans cette modélisation, chaque terme est représenté sous forme de multigraphes orientés, les sommets représentant un nom. Dans ce modèle le concept de restriction est facilement capturé : la portée d'un nom devenant sa composante connexe

[2] L'équivalence par permutations est basée sur le concept d'indépendance, et donc sur la notion de causalité entre réduction.

avait soulevé de nombreuses questions dans la dernière partie de ce rapport. D'autres développements sont également a envisagé. Il serait par exemple intéressant de trouver une structure d'effet pour notre structure d'action. Mais pour terminer cette conclusion sur une remarque d'ordre plus personnelle, j'aimerais dire que ce travail m'aura permis de découvrir, et de me confronter, aux problèmes de la sémantique des languages concurrents. sujet au carrefour de trois des grands défis posés à la recherche en informatique : concevoir et maintenir des logiciels sûrs et fiables, maîtriser l'informatique distribuée et programmer les machines parallèles. J'espère d'ailleurs vous avoir communiquer, au travers des différentes remarques de ce rapport, mon enthousiasme ainsi que ma confiance dans le domaine fructueux de la sémantique.

Annexe A

Sémantiques opérationnelles.

		$x(y).P \xrightarrow{x(w)} P\{w/y\}$	
		$\bar{x}w.P \xrightarrow{\bar{x}w} P$	
$P \xrightarrow{\alpha} P'$	$\vdash$	$!P \xrightarrow{\alpha} P' \parallel !P$	
	$\vdash$	$P \parallel Q \xrightarrow{\alpha} P' \parallel Q$	$\mathrm{bn}(\alpha) \cap \mathrm{fn}(Q) = \emptyset$
	$\vdash$	$(\nu\ x)\ P \xrightarrow{\alpha} (\nu\ x)\ P'$	$x \notin \mathrm{names}(\alpha)$
Communication:			
$P \xrightarrow{x(w)} P'$ $Q \xrightarrow{\bar{x}w} Q'$	$\vdash$	$P \parallel Q \xrightarrow{\tau} P' \parallel Q'$	
Open:			
$P \xrightarrow{\bar{x}w} P'$	$\vdash$	$(\nu\ w)\ P \xrightarrow{\bar{x}(y)} P'\{y/w\}$	$x \neq w$ et $y \notin\ \mathrm{fn}((\nu w)P)$
Close:			
$P \xrightarrow{x(y)} P'$ $Q \xrightarrow{\bar{x}(y)} Q'$	$\vdash$	$P \parallel Q \xrightarrow{\tau} (\nu\ y)(P' \parallel Q')$	

TAB. A.1 - système de transitions avec préinstantiation : EPIC.

		$x(y).P \xrightarrow{x(w)} P\{w/y\}$	
		$\bar{x}w.P \xrightarrow{\bar{x}w} P$	
$P \xrightarrow{\theta} P'$	$\vdash$	$!P \xrightarrow{\theta} P' \parallel !P$	
	$\vdash$	$P \parallel Q \xrightarrow{\theta \;\mid\; id} P' \parallel Q$	$\mathrm{bn}(\theta) \cap \mathrm{fn}(Q) = \emptyset$
	$\vdash$	$(\nu\ x)\ P \xrightarrow{(x)\theta} (\nu\ x)\ P'$	$x \notin \mathrm{names}(\theta)$
Communication: $P \xrightarrow{\theta_1} P'$ $Q \xrightarrow{\theta_2} Q'$ $\mathrm{cl}(\theta_1) = \overline{\mathrm{cl}(\theta_2)}$ $\mathrm{vl}(\theta_1) = \mathrm{vl}(\theta_2)$	$\vdash$	$P \parallel Q \xrightarrow{\theta_1 \;\mid\; \theta_2} P' \parallel Q'$	

TAB. A.2 - système de transitions prouvées pour EPIC (1).

Open:

$$P \xrightarrow{\theta} P' \qquad \vdash \qquad (\nu\ x)\ P \xrightarrow{y \,\triangleright\, (\square\ \theta\{y/x\})} P'\{y/x\}$$

$$x \neq w \text{ et } y \notin \ \mathrm{fn}((\nu w)P)$$

$\mathrm{vl}(\theta) = x$
$\mathrm{cl}(\theta) \neq x$

Propagate (par):

$$P \xrightarrow{y \,\triangleright\, \theta} P' \qquad \vdash \qquad P \parallel Q \xrightarrow{y \,\triangleright\, (\theta \ \mid \ id)} P' \parallel Q \qquad y \notin \ \mathrm{fn}(Q)$$

Propagate (new):

$$\vdash \qquad (\nu\ x)\ P \xrightarrow{y \,\triangleright\, ((x)\theta)} (\nu\ x)\ P'$$

Close:

$$P \xrightarrow{\theta_1} P'$$

$$Q \xrightarrow{y \,\triangleright\, \theta_2} Q' \qquad \vdash \qquad P \parallel Q \xrightarrow{(\triangledown y)(\theta_1 \ \mid \ \theta_2)} (\nu\ y)(P' \parallel Q')$$

$\mathrm{cl}(\theta_1) = \overline{\mathrm{cl}(\theta_2)}$
$\mathrm{bl}(\theta_1) = y$

TAB. A.3 - système de transitions prouvées pour EPIC (2).

		$x(y).P \xrightarrow{x} (y).P$	
		$\bar{x}w.P \xrightarrow{\bar{x}} [w].P$	
$P \xrightarrow{\alpha} P'$	$\vdash$	$!P \xrightarrow{\alpha} P' \parallel !P$	
	$\vdash$	$P \parallel Q \xrightarrow{\alpha} P' \parallel Q$	
	$\vdash$	$(\nu\ x)\ P \xrightarrow{\alpha} (\nu\ x)\ P'$	$\alpha \notin \{x, \bar{x}\}$
	$\vdash$	$(y).P \xrightarrow{\alpha} (y)\ P'$	
	$\vdash$	$[w].P \xrightarrow{\alpha} [w].P'$	
Communication: $P \xrightarrow{x} P'$ $Q \xrightarrow{\bar{x}} Q'$	$\vdash$	$P \parallel Q \xrightarrow{\tau} P' \bullet Q'$	
Commitment:		$((x).P) \bullet ([y].Q) \longrightarrow P\{y/x\} \parallel Q$	
Distribute (par): $P \bullet Q \longrightarrow R$	$\vdash$	$(P \parallel S) \bullet Q \longrightarrow R \parallel S$ $P \bullet (S \parallel Q) \longrightarrow S \parallel R$ $\ldots$	
Distribute (new): $x \notin \mathrm{fn}(Q)$ $y \notin \mathrm{fn}(P)$	$\vdash$	$((\nu\ y)\ P\{y/x\}) \bullet Q \longrightarrow (\nu\ x)\ R$	

TAB. A.4 - système de transitions avec postinstantiation : LPIC.

		$x(y).P \xrightarrow{x} (y).P$	
		$\bar{x}w.P \xrightarrow{\bar{x}w} [w].P$	
$P \xrightarrow{\theta} P'$	$\vdash$	$!P \xrightarrow{\theta} P' \parallel !P$	
	$\vdash$	$P \parallel Q \xrightarrow{\theta \mid id} P' \parallel Q$	
	$\vdash$	$(\nu\ x)\ P \xrightarrow{(x)\ \theta} (\nu\ x)\ P'$	$l(\theta) \notin \{x, \bar{x}\}$
	$\vdash$	$(y).P \xrightarrow{\theta} (y).P'$	
	$\vdash$	$[w].P \xrightarrow{\theta} [w].P'$	
Communication: $P \xrightarrow{\theta_1} P'$ $Q \xrightarrow{\theta_2} Q'$ $l(\theta_1) = \overline{l(\theta_2)}$	$\vdash$	$P \parallel Q \xrightarrow{\theta_1 \mid \theta_2} P' \bullet Q'$	
Commitment:		$((x).P) \bullet ([y].Q) \longrightarrow P\{y/x\} \parallel Q$	

TAB. A.5 - système de transitions prouvées pour LPIC (1).

Open:		
$P \xrightarrow{\theta} P'$	$\vdash$	$(\nu\ x)\ P \xrightarrow{x \triangleright (\Box\ \theta)} P'$
$\mathrm{vl}(\theta) = x$		
$\mathrm{cl}(\theta) \neq x$		
Propagate (par):		
$P \xrightarrow{x \triangleright \theta} P'$	$\vdash$	$P \parallel Q \xrightarrow{y \triangleright (\theta\{y/x\} \ \mid \ id)} P'\{y/x\} \parallel Q$
		$y \notin \ \mathrm{fn}(P \cup Q)$
Propagate (new):		
	$\vdash$	$(\nu\ z)\ P \xrightarrow{y \triangleright ((z)\theta\{y/x\})} (\nu\ z)\ P'\{y/x\}$
Close:		
$P \xrightarrow{\theta_1} P'$		
$Q \xrightarrow{x \triangleright \theta_2} Q'$	$\vdash$	$P \parallel Q \xrightarrow{(\nabla y)(\theta_1 \ \mid \ \theta_2\{y/x\})} (\nu\ y)(P' \bullet Q'\{y/x\})$
$\mathrm{cl}(\theta_1) = \overline{\mathrm{cl}(\theta_2)}$		
$y \notin \ \mathrm{fn}(P \cup Q)$		
Distribute (par):		
$P \bullet Q \to (R_1 \parallel R_2)$	$\vdash$	$(P \parallel S) \bullet Q \longrightarrow (R_1 \parallel S) \parallel R_2$
		$P \bullet (S \parallel Q) \longrightarrow R_1 \parallel (S \parallel R_2)$
		$\ldots$
Distribute (new):		
$y \notin \ \mathrm{fn}(P)$	$\vdash$	$((\nu\ y)\ P\{y/x\}) \bullet Q \longrightarrow ((\nu\ x)\ R_1) \parallel R_2$

TAB. A.6 - système de transitions prouvées pour LPIC (2).

$$x(y).P \xrightarrow{x(y)} (y).P$$

$$\bar{x}w.P \xrightarrow{\bar{x}w} [w].P$$

$P \xrightarrow{\theta} P'$	$\vdash$	$!P \xrightarrow{\theta} P' \parallel !P$	
	$\vdash$	$P \parallel Q \xrightarrow{\theta \ \mid \ id} P' \parallel Q$	
	$\vdash$	$(\nu\ x)\ P \xrightarrow{(x)\ \theta} (\nu\ x)\ P'$	$cl(\theta) \notin \{x, \bar{x}\}$
	$\vdash$	$(y).P \xrightarrow{\theta} (y).P'$	
	$\vdash$	$[w].P \xrightarrow{\theta} [w].P'$	

Communication:

$P \xrightarrow{\theta} P'$ $Q \xrightarrow{\bar{\theta}} Q'$ $P' \bullet Q' \xrightarrow{e} R$	$\vdash$	$P \parallel Q \xrightarrow{e} R$

TAB. A.7 - FREEPIC (1).

Commitment:

$$((x).P) \bullet ([y].Q) \xrightarrow{\{y/x\} \ | \ \bullet} P\{y/x\} \parallel Q$$

Distribute :

$$P \bullet Q \xrightarrow{\theta_1 \ | \ \theta_2} (R_1 \parallel R_2) \quad \vdash \quad (P \parallel S) \bullet Q \xrightarrow{(\theta_1 \ | \ id) \ | \ \theta_2} (R_1 \parallel S) \parallel R_2$$

$$P \bullet (S \parallel Q) \xrightarrow{\theta_1 \ | \ (id \ | \ \theta_2)} R_1 \parallel (S \parallel R_2)$$

$$\ldots$$

$$\vdash \quad ((\nu\ y)\ P\{y/x\}) \bullet Q \xrightarrow{(x)\theta_1 \ | \ \theta_2} ((\nu\ x)\ R_1) \parallel R_2 \qquad y \notin \mathrm{fn}(P)$$

ScopeMigration :

$$(\nu\ x)\ P \xrightarrow{y \triangleright \Box} P\{y/x\} \qquad y \notin \mathrm{fn}(P)$$

ScopeClosure :

$$P \xrightarrow{x \triangleright \theta} P' \quad \vdash \quad P \xrightarrow{(\triangledown x)\ \theta} (\nu\ x)\ P'$$

Propagate :

$$P \xrightarrow{x \triangleright \theta} P' \quad \vdash \quad P \parallel Q \xrightarrow{y \triangleright (\theta\{y/x\} \ | \ id)} P'\{y/x\} \parallel Q \qquad y \notin \mathrm{fn}(P \cup Q)$$

$$\vdash \quad (\nu\ z)\ P \xrightarrow{y \triangleright ((z)\theta\{y/x\})} (\nu\ z)\ P'\{y/x\}$$

TAB. A.8 - FREEPIC (2).

Bibliographie

[1] Banâtre J.P., Métayer D., *The Gamma model and its discipline of programming* , Science of Computer Programming, Vol 15,pp55-77, 1990

[2] Berry G., Lévy J.-J., *Minimal and Optimal Computations of Recursive Programs* , JACM 26, pp148-175, 1979

[3] Berry G., Boudol G., *The Chemical abstract machine* , Journal of Theoretical Computer Science, Vol 96, pp217-248, 1992

[4] Boudol G., *Some Chemical abstract machines* , (à paraître)

[5] Boudol G., *Asynchrony and the π-calculus* , INRIA Res. Report 1702, 1992

[6] Boudol G., Castellani I., *Concurrency and Atomicity* , Theoretical Computer Science 59, pp25-84, 1988

[7] Boudol G., Castellani I., *A non-interleaving semantics for CCS based on proved transitions* , Fundamenta Informaticae XI, pp433-452, 1988

[8] Boudol G., Castellani I., *Flow models of distributed computations: Three equivalent semantics for CCS* , Research Report INRIA 1484, 1991

[9] Boudol G., Castellani I., *Flow models of distributed computations: event structures and nets* , Research Report INRIA 1482, 1991

[10] Mazurkiewicz, *Traces, Histories, Graphs: instances of a process monoid*, Poc. MFCS'84, Lecture Notes in Computer Science 176, pp115-133, 1984

[11] Milner R., *Functions as Processes* , Math. Struct. in Comp. Science, Vol 2, pp119-141, 1992

[12] Milner R., *Calculi for synchrony and asynchrony* , Journal of Theoretical Computer Science, Vol 25, pp267-310, 1983

[13] Milner R., *Action Structures* , Research Report LFCS-92-249, Laboratory for Foundations of Computer Science, Computer Science Department, Edinburgh University, 1992

[14] Milner R., *An action Structure for Synchronous π-calculus* , FCT 93, Lecture Notes in Computer Science 710, pp87-105, 1993

[15] Milner R., *The polyadic π-calculus: a tutorial* , Research Report LFCS-91-180, Laboratory for Foundations of Computer Science, Computer Science Department, Edinburgh University, 1992

[16] Milner R., *Action Calculi or concrete action structures* , Proc. MFCS Conference, Gdansk, Lecture Notes in Computer Science 711,pp105-121, 1993

[17] Milner R., Parrow J., Walker D., *Modal Logics for Mobile Processes* , Research Report LFCS-91-136, Laboratory for Foundations of Computer Science, Computer Science Department, Edinburgh University, 1991

[18] Milner R., Parrow J., Walker D., *A calculus of mobile processes, Parts I and II* , Journal of Information and Computation, Vol 100, pp1-40 & 41-77, 1992

[19] Pierce B., Sangiorgi D., *Typing and Subtyping for Mobile Processes* , 1992

[20] Walker D.,*π-calculus Semantics of Object-Oriented Programming Languages* , Proc. TACS'91. Springer-Verlag LNCS 526, pp532-547, 1991

Table des matières

1 Π-calcul. **2**

2 Structures d'actions. **5**
2.1 Définition. 5
2.2 Effets et structures d'effets. 8
2.3 Structures de processus. 9
2.4 Dynamique des structures de processus. 10
2.5 PIC: une structure d'action pour le π-calcul. 13

3 Équivalence par permutation. **16**
3.1 Système de transitions prouvées pour le π-calcul. 16
3.1.1 Transitions avec préinstantiation (EPIC). 16
3.1.2 Transitions prouvées pour EPIC. 17
3.1.3 Transitions avec postinstantiation (LPIC). 20
3.2 Permutation de transitions concurrentes 20
3.3 Réduction dans $\mathcal{T}(P)$. 26
3.3.1 Réduction de deux transitions concurrentes. 26
3.3.2 Réduction de séquences de transitions prouvées, et problèmes. 28

4 Étude comparée des deux modèles. **32**
4.1 Étude de FREEPIC 32
4.1.1 Développement sur la restriction : la classe de transitions ν 32
4.1.2 Transitions dans FREEPIC 33
4.2 Actions = transitions prouvées. 34
4.2.1 Rapport avec les structures d'action. 37
4.2.2 Relation de réduction sur les actions. 38

5 Conclusion **42**

Annexe **44**

A Sémantiques opérationnelles. **44**