Définition des stratégies paresseuse et stricte pour TPC et TPC_{ES}

Julien Forest

Rapport de stage de DEA

Responsable de projet: Delia Kesner

Table des matières

1 Introduction

1.1 Motivations

L'une des explications possibles du succès toujours croissant des langages fonctionnels est sans aucun doute la possibilité de définir des fonctions par cas. Cette possibilité permet au programmeur de mélanger à la fois les manières extensionnelle (association explicite d'un résultat à un argument) et intentionnelle (description d'une méthode de calcul), de définir une fonction. Elle permet de plus d'écrire des programmes en général plus clairs donc plus faciles à maintenir. Les deux définitions suivantes de la fonction merge (qui prend en argument deux listes et renvoie une liste formée des éléments interclassés de ces deux listes) illustrent bien ce propos:

```
#let rec merge = function
 ([]        , l)    -> l
|(a::l as z , [])   -> z
|(a::l      , b::m) -> a::(b::merge(l,m))
;;

#let rec merge = function (l,m) ->
if l=[] then m
  else if m=[] then l
      else h(l)::(h(m)::merge(t(l),t(m)))
;;
```

Le premier programme utilise les deux grands avantages de la programmation fonctionnelle: le filtrage (qui donne la possibilité de définir les fonctions par cas) et l'utilisation des motifs. Chaque ligne définit un cas de filtrage à travers les motifs qui permettent de donner la forme que doit avoir un argument (a::l signifie que l'argument est une liste avec au moins un élément appelé a). L'avantage du premier programme sur le second est sa plus grande clarté qui permet de mieux le comprendre et de l'écrire plus facilement. Dans le second programme la fonction h renvoie le premier élément d'une liste, la fonction t renvoie la liste privée de son premier élément.

Le λ-calcul classique, introduit par Church [5, 6], est souvent considéré comme le modèle des langages de programmation fonctionnels (tels que SML [15], HASKELL [11], CAML [14]) bien qu' il ne permette ni les définitions par cas ni le traitement des motifs. Plusieurs calculs ont été mis au point pour rendre compte de ces deux possibilités des langages fonctionnels [13, 4]. En particulier, Tannen, Kesner et Puel ont présenté [13] un calcul avec substitution et filtrage implicites, c'est à dire un calcul où les opérations de filtrage et de substitutions sont définies en dehors du calcul par des fonctions situées dans un meta-niveau.
Un autre inconvénient majeur du λ-calcul classique est qu' il traite l'opération de substitution de manière implicite (dans un meta-niveau). Afin de pallier à

ce problème et de se rapprocher du niveau réel de l'implantation, divers dérivés du λ-calcul traitant la substitution de manière explicite ont été mis au point notamment par Abadi, Cardelli, Curien et Levy [1] ou par [2, 3, 12].
Dans le cadre des calculs avec motifs, Kesner et Cerrito ont mis au point [4] deux nouveaux calculs TPC et TPC_{ES} qui sont tous les deux des calculs avec filtrage explicite. Dans TPC la substitution est traitée à un meta-niveau alors que dans TPC_{ES} elle est traitée dans le langage lui même.

Le but de ce stage était de définir deux stratégies de calcul pour chacun de ces deux calculs: une stratégie paresseuse et une stratégie stricte. Les stratégies paresseuses n'effectuent un calcul que quand celui-ci est nécessaire alors que les stratégies strictes effectuent tous les calculs possibles dans les arguments d'une fonction avant d'effectuer des calculs sur l'appel de la fonction elle-même. L'un des principaux intérêts de définir des stratégies pour des calculs est de permettre la définition de langages de programmation qui implantent toujours une stratégie particulière d'un calcul.

Nous nous sommes beaucoup inspiré de l'article de Tannen, Kesner et Puel [13] pour trouver les idées nécessaires à une telle définition.

Dans la suite de cette introduction, nous donnons une définition formelle du λ-calcul, de l'isomorphisme de Curry-Howard, de la notion de substitution explicite, du calcul des séquents et de TPC_{ES} et TPC. Enfin nous définirons les notions de stratégie et d'évaluateur en vue de les appliquer dans les sections suivantes.

1.2 Terminologie

Nous allons à présent donner rapidement quelques définitions dont nous nous servirons plus loin.

Définition 1.1

- Un *système de réduction* est la donnée d'un couple ($\mathcal{A}$,$\rightarrow$) où $\mathcal{A}$ est un ensemble et $\rightarrow$ est une relation de $\mathcal{A}$ dans lui-même.

- La *clôture transitive* d'une relation de réduction $\rightarrow$ est la relation $\rightarrow^+$ définie sur le même ensemble que $\rightarrow$ et telle que: t$\rightarrow^+$ t' si et seulement si il existe un entier n>0 et il existe $t_1 = t, t_2, \ldots, t_{n-1}, t_n = t'$ tels que $t_1 \rightarrow t_2 \rightarrow \ldots \rightarrow t_n$.

- La *clôture transitive et réflexive* d'une relation de réduction $\rightarrow$ est la relation de réduction $\rightarrow^*$ telle que t $\rightarrow^*$ t' si et seulement si t=t' ou t$\rightarrow^+$t'.

Souvent, plutôt que de définir la relation $\rightarrow$ en donnant explicitement tous les couples en relation par $\rightarrow$, on donne $\rightarrow$ comme un ensemble de règles applicables sous certaines conditions (par exemple la forme du premier élément du couple)

Définition 1.2 On dit qu'une relation de réduction $\rightarrow$ est *confluente* si et seulement si pour tout élément t de $\mathcal{A}$ tel que $t \rightarrow^* t_1$ et $t \rightarrow^* t_2$ il existe t_3 tel que $t_1 \rightarrow^* t_3$ et $t_2 \rightarrow^* t_3$.

Définition 1.3 Soit $(\mathcal{A}, \rightarrow)$ un système de réduction.

- Un élément t de $\mathcal{A}$ est dit être *une forme normale* pour $\rightarrow$ si t n'est en relation par $\rightarrow$ avec aucun autre élément de $\mathcal{A}$.
- On dit que t' est une forme normale de t si t est une forme normale pour $\rightarrow$ et si t $\rightarrow^*$ t'.
- On dit qu'une relation de réduction $\rightarrow$ *termine* sur $\mathcal{A}$ si tout élément de cet ensemble possède une forme normale pour $\rightarrow$.
- On dit qu'une relation de réduction $\rightarrow$ est *fortement normalisante* sur $\mathcal{A}$ si il n'existe pas de séquence infinie de réduction $t_0 \rightarrow t_1 \rightarrow \ldots$.

Remarque 1.1 Il est clair qu'une relation de réduction $\rightarrow$ qui termine et est confluente est fortement normalisante.

Définition 1.4

- Une *règle d'inférence* d'un système de preuve est un couple formé d'un ensemble d'expressions (les *prémisses*) et d'une expression (la *conclusion*). On a l'habitude de noter une règle d'inférence avec prémisses $A_1, \ldots, A_n$ et conclusion B sous la forme: $\frac{A_1 \quad \ldots \quad A_n}{B}$.
- Une règle d'inférence sans prémisses est un *axiome*.

Définition 1.5 Un *séquent* de la logique propositionelle est un couple formé d'un multi-ensemble de formules de la logique propositionelle Γ (l'*antécédent*) et d'une formule de la logique propositionelle A le (*conséquent*). Un tel séquent est en général noté $\Gamma \vdash A$.

Intuitivement le séquent $\Gamma \vdash A$ peut s'interpréter comme signifiant "Sous les hypothèses Γ on peut déduire A".

1.3 La déduction naturelle et le λ-calcul

1.3.1 Le λ-calcul

Nous allons maintenant brièvement introduire le λ-calcul [5].
Les termes du λ-calcul sont donnés par la grammaire suivante:

$$M\ :\ :\ = x \mid \lambda x.M \mid (M\ M)$$

où x appartient à un ensemble infini de *variables* notées $x, y, z, t, \ldots$ Le terme $\lambda x.M$ est une λ-*abstraction* et $(M\ M)$ est une *application*.

On définit les *variables libres* d'un terme par induction sur la structure de ce terme comme suit:

$$\begin{array}{lll} FV(x) & = & \{x\} \\ FV(\lambda x.M) & = & FV(M) \setminus \{x\} \\ FV(M\ N) & = & FV(M) \cup FV(N) \end{array}$$

On définit les *variables liées* d'un terme par induction sur la structure de ce terme comme suit:

$$\begin{array}{lll} BV(x) & = & \emptyset \\ BV(\lambda x.M) & = & BV(M) \cup \{x\} \\ BV(M\ N) & = & BV(M) \cup BV(N) \end{array}$$

Exemple 1.2

- Soit t=$\lambda x.(x\ y)$ on a: FV(t)=$\{y\}$ et BV(t)=$\{x\}$.
- Soit t'=$((\lambda x.x)\ x)$ on: FV(t')=$\{x\}$ et BV(t)=$\{x\}$.

On dit que deux termes sont *α-équivalents* si ils ne diffèrent que par le nom de leurs variables liées. Par exemple, $\lambda x.x$ et $\lambda y.y$ sont α-équivalents, en revanche $\lambda x.x$ et $\lambda x.y$ ne le sont pas. Dans la suite, on ne considérera que des termes t tels que $BV(t) \cap FV(t) = \emptyset$. Ceci nous sera toujours possible puisque si t ne vérifie pas cette condition il existe toujours un terme t' qui lui est α-équivalent et qui la vérifie.

Le λ-calcul classique ne comprend qu'une règle de réduction:

$$(\lambda x.M\ N) \rightarrow_\beta M\{x \leftarrow N\}$$

la notation $M\{x \leftarrow N\}$ signifie que dans M toutes les occurrences libres de x ont été remplacées par N. Cette opération est appelée *substitution implicite*. C'est une opération situé dans un meta-niveau par rapport au λ-calcul en ce sens qu'elle ne fait pas partie du calcul en lui-même (aucune règle de calcul ne définit cette opération).

Le λ-calcul classique est confluent mais il ne termine pas. Par exemple, le terme $\Omega = (\lambda x.(x\ x)\ \lambda y.(y\ y))$ n'a pas de forme normale puisqu'il se réduit uniquement en lui-même. Ce calcul a la puissance des machines de Turing c'est à dire qu'il peut coder n'importe quel programme exécutable par une machine Turing.

1.3.2 Le λ-calcul typé

On peut définir un sous calcul du λ-calcul qui possède à la fois les propriétés de confluence et de terminaison: le λ-calcul simplement typé. Ce calcul est fondé sur la même grammaire et sur la même règle de réduction que le λ-calcul mais l'ensemble des termes de ce calcul est restreint aux termes *typables* du λ-calcul. Un terme t du λ-calcul est dit *typable* si on peut lui affecter un type grâce au système de typage suivant.

On définit l'ensemble des *types* du λ-calcul comme étant les termes donnés par la grammaire suivante:

$$\sigma ::= \iota \mid \sigma \to \sigma$$

où ι est un ensemble de types prédéfinis.

Les règles de typages du λ-calcul sont les suivantes:

$$\frac{}{x{:}A,\Gamma\vdash x{:}A}\ (\mathrm{Ax}) \qquad \frac{x{:}A,\Gamma\vdash M{:}B}{\Gamma\vdash\lambda x.M{:}A\to B}\ (\mathrm{Abs}) \qquad \frac{\Gamma\vdash M{:}A\to B \qquad \Gamma\vdash N{:}A}{\Gamma\vdash(M\ N){:}B}\ (\mathrm{App})$$

La notation $\Gamma \vdash$ M:A se lit "Dans l'environnement Γ, le terme M a le type A".

Ces règles de typages sont calquées sur les règles d'inférence du fragment du système de preuve en logique propositionelle dit de la *déduction naturelle* dont les règles d'inférences sont les suivantes:

$$\frac{}{A_1,\ldots,A_n\vdash A_i}\prec \text{ Si } 1\leq i\leq n \qquad \frac{A,\Gamma\vdash B}{\Gamma\vdash A\Rightarrow B} \qquad \frac{\Gamma\vdash A\Rightarrow B \qquad \Gamma\vdash A}{\Gamma\vdash B}$$

Exemple 1.3

- Le terme $\lambda x.\lambda y.x$ est typable comme le montre l'arbre de typage suivant:

$$\dfrac{\dfrac{\dfrac{}{y{:}\beta,x{:}\alpha\vdash x{:}\alpha}(Ax)}{x{:}\alpha\vdash\lambda y.x{:}\beta\to\alpha}(Abs)}{\vdash\lambda x.\lambda y.x{:}\alpha\to(\beta\to\alpha)}\ (\mathrm{Abs})$$

- En revanche le terme $\lambda x.(x\ x)$ n'est pas typable. On suppose en effet que ce terme possède un type β, on peut alors construire l'arbre suivant:

$$\dfrac{\dfrac{x{:}\alpha\vdash x{:}\gamma\to\beta \qquad x{:}\alpha\vdash x{:}\beta}{x{:}\alpha\vdash(x\ x){:}\beta}(App)}{\vdash\lambda x.(x\ x)}\ (\mathrm{Abs})$$

  On doit alors avoir $\beta = \alpha = \gamma \to \beta$ ce qui est clairement impossible.

Dans les deux exemples ci-dessus nous avons utilisé les notations α et β plutôt que A et B afin d'insister sur le fait que les preuves que nous donnons sont valables quelque soit la valeur de α et β (que l'on peut donc considérer comme des *variables de type*).

1.3.3 L'isomorphisme de Curry-Howard

Il existe un isomorphisme évident du à Curry et Howard entre les preuves des théorèmes de la déduction naturelle et les dérivations de type du λ-calcul typé. A chaque type du λ-calcul typé correspond une proposition de la déduction naturelle. De plus à chaque preuve correspond un terme et inversement. Cet isomorphisme peut se résumer ainsi:

λ-calcul	$\Leftrightarrow$	logique propositionelle en déduction naturelle
types	$\Leftrightarrow$	propositions
termes	$\Leftrightarrow$	preuves
réduction	$\Leftrightarrow$	normalisation de preuves
types de base: A,B,C,...	$\Leftrightarrow$	Propositions atomiques A,B,C,...

Dans le cadre de cet isomorphisme l'axiome $\frac{}{x:A,\Gamma\vdash x:A}$ des règles de typage du λ-calcul correspond à l'axiome $\frac{}{A_1,\ldots,A_n\vdash A_i}$ de la déduction naturelle en logique propositionelle, la règle de typage $\frac{x:A,\Gamma\vdash M:B}{\Gamma\vdash\lambda x.M:A\to B}$ correspond à la règle de la déduction naturelle $\frac{A,\Gamma\vdash B}{\Gamma\vdash A\Rightarrow B}$ et enfin la règle de typage $\frac{\Gamma\vdash M:A\to B \qquad \Gamma\vdash N:A}{\Gamma\vdash(M\ N):B}$ correspond à la règle de la déduction naturelle $\frac{\Gamma\vdash A\Rightarrow B \qquad \Gamma\vdash A}{\Gamma\vdash B}$

1.3.4 Extensions du λ-calcul simplement typé

On peut étendre la déduction naturelle aux connecteurs $\vee$ et $\wedge$ par les règles d' inférence suivantes:

$$\frac{\Gamma\vdash A}{\Gamma\vdash A\vee B}\ (\vee intro1) \qquad \frac{\Gamma\vdash B}{\Gamma\vdash A\vee B}\ (\vee intro2) \qquad \frac{\Gamma\vdash A\vee B \qquad A,\Gamma\vdash C \qquad B,\Gamma\vdash C}{\Gamma\vdash C}\ (\vee elim)$$

$$\frac{\Gamma\vdash A\wedge B}{\Gamma\vdash A}\ (\wedge elim1) \qquad \frac{\Gamma\vdash A\wedge B}{\Gamma\vdash B}\ (\wedge elim2) \qquad \frac{\Gamma\vdash A \qquad \Gamma\vdash B}{\Gamma\vdash A\wedge B}\ (\wedge intro)$$

On remarque qu'en déduction naturelle chacun des trois connecteurs ($\Rightarrow$, $\vee$ et $\wedge$) possède une (ou deux) règle d'introduction et une (ou deux) règle d'élimination.

Dans l'esprit de l'isomorphisme de Curry-Howard on peut alors étendre le λ-calcul ajoutant à sa grammaire les termes suivants:

$$\begin{array}{lll} M & ::= & \texttt{inr}(M) \\ & & \mid \texttt{inl}(M) \\ & & \mid \langle M, M\rangle \\ & & \mid \pi_i(M) \\ & & \mid \text{case } M \text{ of } x : A.M \mid y : B.M \end{array}$$

où $1 \leq i \leq 2$, $\texttt{inr}(M)$ et $\texttt{inl}(M)$ sont respectivement les *injections droite* et *gauche*, $\langle M, M\rangle$ est une *paire*, π_i(M) est une *projection*, et case M of $x : A.M \mid y : B.M$ est une *définition par cas.*

Un terme de la forme case M of $x : A.N \mid y : B.J$ est appelé définition par cas pour la raison qu'un tel terme dépend fortement de la forme que prendra M au cours des calculs (suivant les *cas* on poursuivra avec N ou J) comme le montreront les règles de réduction ci-dessous.

La grammaire des types devient alors la suivante:

$$\sigma ::= \iota \mid \sigma \to \sigma \mid \sigma \times \sigma \mid \sigma + \sigma$$

où $\sigma \times \sigma$ est un *type produit* et $\sigma + \sigma$ un *type somme.*

Les règles de typages s'enrichissent alors des règles suivantes:

$$\frac{\Gamma \vdash M:B}{\Gamma \vdash \mathtt{inr}_A(M):A+B}\ (+intro_1) \qquad \frac{\Gamma \vdash M:A}{\Gamma \vdash \mathtt{inl}_B(M):A+B}\ (+intro_2)$$

$$\frac{\Gamma \vdash L:A+B \qquad x:A,\Gamma \vdash M:C \qquad y:B,\Gamma \vdash N:C}{\Gamma \vdash (\text{case } L \text{ of } x:A.M|y:B.N):C}\ (+elim)$$

$$\frac{\Gamma \vdash M:A \qquad \Gamma \vdash N:B}{\Gamma \vdash \langle M,N\rangle:A \times B}\ (\times intro)$$

$$\frac{\Gamma \vdash M:A_1 \times A_2}{\Gamma \vdash \pi_i(M):A_i}\ (\times elim_i)$$

L'isomorphisme de Curry-Howard peut alors s'étendre comme suit:

$\frac{\Gamma \vdash M:A}{\Gamma \vdash \mathtt{inr}_B(M):A+B}\ (+intro_1)$	$\Leftrightarrow$	$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B}$
$\frac{\Gamma \vdash M:B}{\Gamma \vdash \mathtt{inl}_A(M):A+B}\ (+intro_2)$	$\Leftrightarrow$	$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B}$
$\frac{\Gamma \vdash L:A+B \qquad x:A,\Gamma \vdash M:C \qquad y:B,\Gamma \vdash N:C}{\Gamma \vdash (\text{case } L \text{ of } x:A.M\|y:B.N):C}\ (+elim)$	$\Leftrightarrow$	$\frac{\Gamma \vdash A \vee B \qquad A,\Gamma \vdash C \qquad B,\Gamma \vdash C}{\Gamma \vdash C}$
$\frac{\Gamma \vdash M:A \qquad \Gamma \vdash N:B}{\Gamma \vdash \langle M,N\rangle:A \times B}\ (\times intro)$	$\Leftrightarrow$	$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$
$\frac{\Gamma \vdash M:A_1 \times A_2}{\Gamma \vdash \pi_i(M):A_i}\ (\times elim_i)$	$\Leftrightarrow$	$\frac{\Gamma \vdash A_1 \times A_2}{\Gamma \vdash A_i}$

On ajoute enfin aux règles de réduction du λ-calcul classique les règles suivantes:

$$\begin{array}{lll}
\text{case } \mathtt{inl}(L) \text{ of } x.M \mid y.N & \to_{Case1} & M\{x \leftarrow L\} \\
\text{case } \mathtt{inr}(L) \text{ of } x.M \mid y.N & \to_{Case2} & N\{x \leftarrow L\} \\
\pi_i(\langle M_1, M_2\rangle) & \to_{\pi_i} & M_i
\end{array}$$

1.3.5 La substitution explicite

Un rapprochement possible du λ calcul classique par rapport à l'implantation est le remplacement de la meta-opération de substitution implicite par une opération de substitution explicite [1, 2, 3, 12]. On peut étendre la grammaire du λ-calcul en lui rajoutant les termes de la forme $M[x/M]$ (les *substitutions explicites*) et en remplaçant sa règle de réduction par les règles suivantes:

$$\begin{array}{lll}
(\lambda x.M\ N) & \to_{Beta} & M[x/N] \\
x[x/N] & \to_{sub1} & N \\
y[x/N] & \to_{sub2} & y \text{ si } y \neq x \\
\lambda x.M[y/N] & \to_{sub3} & \lambda x.M[y/N] \\
(M\ N)[x/L] & \to_{sub4} & (M[x/L]\ N[x/L])
\end{array}$$

On notera $M \rightarrow_{\lambda_x} M_1$ pour exprimer le fait que m se réduit en M_1 par l'une des règles de réduction précédentes.

Les règles de typage sont les suivantes:

$$\frac{}{x{:}\alpha,\Gamma\vdash x{:}\alpha}\ (\mathrm{Ax}) \qquad\qquad \frac{x{:}\alpha,\Gamma\vdash M{:}\beta}{\Gamma\vdash\lambda x.M{:}\alpha\rightarrow\beta}\ (\mathrm{Abs})$$

$$\frac{\Gamma\vdash M{:}\alpha\rightarrow\beta \qquad \Gamma\vdash N{:}\alpha}{\Gamma\vdash(M\ \ N){:}\beta}\ (\mathrm{App}) \qquad\qquad \frac{\Gamma\vdash M{:}A \qquad x{:}A,\Gamma\vdash N{:}B}{\Gamma\vdash N[x/M]{:}B}\ (\mathrm{Sub})$$

Le calcul ainsi défini est appelé λ_x calcul introduit par [3].

Dans les règles de typage ci-dessus la règle (Sub) ne correspond à aucune règle de la déduction naturelle. Ceci nous fait perdre la notion d'isomorphisme entre la déduction naturelle et le système de typage du λ-calcul typé.

L'un des avantages majeurs de l'utilisation de la substitution explicite est que l'on se rapproche du niveau réel de l'implantation.

L'une des propriétés du λ_x calcul est qu'il implante bien l'idée naturelle de substitution. Plus formellement on a le résultat suivant [2]:

Proposition 1.4

- Si t est un terme du λ-calcul alors:

  $$t \rightarrow_\beta t' \Rightarrow t \rightarrow^*_{\lambda_x} t'$$

- Si t est un terme du λ_x calcul et si t $\rightarrow_{\lambda_x}$ t' alors X(t)$\rightarrow_\beta$ X(t') où X(t) désigne la forme normale de t pour le sous calcul du λ_x calcul restreint aux seules règles sub_i.

1.4 Le calcul des séquents et TPC_{ES}

1.4.1 Le calcul des séquents

D'autres calculs typés [4, 13, 10] ont été développés sur le principe de l'isomorphisme de Curry-Howard mais en se fondant sur d'autres systèmes de preuve comme le calcul des séquents de Gentzen (que nous allons maintenant introduire dans le cadre de la logique propositionelle) plutôt que sur la déduction naturelle.

Le *calcul des séquents* est un système formel proposé par Gentzen pour spécifier des preuves en logique propositionelle dont les règles d'inférence sont les suivantes:

$$\frac{}{A_1,\ldots,A_n\vdash A_i}\ (\mathrm{Ax})$$

$$\frac{\Gamma\vdash A \qquad A,\Gamma\vdash B}{\Gamma\vdash B}\ (\mathrm{Cut})$$

$\frac{A,A,\Gamma\vdash B}{A,\Gamma\vdash B}$ (Contraction)

$\frac{\Gamma\vdash B}{A,\Gamma\vdash B}$ (Weakening)

$\frac{\Gamma\vdash A \qquad \Gamma\vdash B}{\Gamma\vdash A\wedge B}$ ($\wedge$right) $\qquad$ $\frac{A,B,\Gamma\vdash C}{A\wedge B,\Gamma\vdash C}$ ($\wedge$left)

$\frac{\Gamma\vdash A}{\Gamma\vdash A\vee B}$ ($\vee$right$_1$) $\quad$ $\frac{\Gamma\vdash B}{\Gamma\vdash A\vee B}$ ($\vee$right$_2$) $\qquad$ $\frac{A,\Gamma\vdash C \qquad B,\Gamma\vdash C}{A\vee B,\Gamma\vdash C}$ ($\vee$left)

$\frac{A,\Gamma\vdash B}{\Gamma\vdash A\Rightarrow B}$ ($\Rightarrow$right) $\qquad$ $\frac{\Gamma\vdash A \qquad B,\Gamma\vdash C}{A\Rightarrow B,\Gamma\vdash C}$ ($\Rightarrow$left)

Ces règles comportent des règles structurelles (les règles (Ax), (Cut), (Contraction) et (Weakening)) ne concernant pas les connecteurs logiques et des règles logiques qui définissent les connecteurs logiques.
Dans ces règles, la règle (Cut) peut s' interpréter comme une règle d'utilisation de lemmes. On peut en effet interpréter cette règle comme: “Si, sous les hypothèses Γ, on peut démontrer le lemme A et, grâce au lemme A et aux hypothèses Γ démontrer le théorème B alors on peut démontrer B sous les hypothèses Γ”.

Contrairement à la déduction naturelle les règles ne sont plus séparées en règles d'introduction et d'élimination des connecteurs mais en règles gauches et règles droites pour ces connecteurs.

1.4.2 Les calculs TPC et TPC_{ES}

En s'inspirant du principe de l'isomorphisme de Curry-Howard, plusieurs calculs typés basés sur le calcul des séquents ont vu le jour. Celui de [13] est un calcul typé avec filtrage et substitution implicite. De même, TPC et TPC_{ES} que nous allons maintenant présenter, sont deux calculs typés dont le système de typage est basé sur le calcul des séquents [4].

Les termes du calcul TPC_{ES} sont définis comme suit.

L'ensemble des *types* pour TPC_{ES} est défini par la grammaire suivante:

$$A ::= \iota \mid A \times A \mid A + A \mid A \to A$$

où ι est un *type de base* (un élément d'un ensemble de types prédéfinis) et les symboles $\times$, $+$ et $\to$ définissent respectivement le produit et la somme de types et les types fonctionnels.
On distingue deux ensembles disjoints de variables: les *variables usuelles* notées x,y,v,z,...et les *variables de somme* $\xi, \rho, \psi, \ldots$. Les variables de somme jouent un rôle essentiel pour les motifs disjonctifs et rendent possible la définition de fonctions par cas.

Les *motifs* (P) sont définis par la grammaire suivante:

$$P ::= \_ \mid x \mid \sharp x \mid \langle P, P\rangle \mid (P \mid_\xi P) \mid @(P, P)$$

Le motif _ est appelé *wildcard*, x est une *variable*, $\#x$ un *motif flèche*, $\langle P, P\rangle$ un *motif produit*, $(P \mid_\xi P)$ un *motif somme* (ou disjonctif) et $@(P, P)$ un *motif de contraction*.

Les *termes* (M) sont définis par la grammaire suivante:

$$
\begin{array}{lcl}
M & ::= & x \mid \langle M, M\rangle \mid \mathtt{inl}_A(M) \mid \mathtt{inr}_A(M) \\
 & & \mid M[x/M] \mid M[\xi/\mathtt{K}] \mid \lambda P\!:\!A.M \mid [M \mid_T M] \\
 & & \mid \mathtt{let}\ M\ \mathtt{be}\ P\!:\!A\ \mathtt{in}\ M \\
 & & \mid z\ \mathtt{of}\ M\ \mathtt{is}\ P\!:\!A\ \mathtt{in}\ M \\
 & & \mid \lambda P\!:\!A.M\ \mathtt{of}\ M\ \mathtt{is}\ P\!:\!A\ \mathtt{in}\ M
\end{array}
$$

où $\mathtt{T} \in \{\xi, \mathtt{L}, \mathtt{R}\}$, $\mathtt{K} \in \{\mathtt{L}, \mathtt{R}\}$, ξ est une variable de somme, x et z sont des variables usuelles et $\mathtt{L}$ et $\mathtt{R}$ sont des constantes spéciales (dont on se servira pour instancier les variables de somme).
Le terme x est appelé une *variable*, $\langle M, M\rangle$ est une *paire*, $\mathtt{inr}_A(M)$ et $\mathtt{inl}_A(M)$ sont respectivement les *injections droite* et *gauche*, $M[x/M]$ et $M[\xi/\mathtt{K}]$ sont respectivement les *substitutions explicites usuelles* et *de somme*, $[M \mid_\mathtt{T} M]$ est une *définition par cas*, $\lambda P : A.M$ est une *λ-abstraction*, $\mathtt{let}\ M\ \mathtt{be}\ P : A\ \mathtt{in}\ M$ est un *let* et enfin les termes de la forme $U\ \mathtt{of}\ M\ \mathtt{is}\ P\ :\ A\ \mathtt{in}\ M$ sont des *applications*.

Remarque 1.5 Pour ne pas alourdir encore les notations, dans la suite il nous arrivera fréquemment d'omettre les types dans les termes qui normalement en comportent.

Notation 1.6 Dans toute la suite on note $[\theta]$ une substitution de la forme [x / M] ou $[\xi/\mathtt{K}]$ et Θ une liste de substitutions $[\theta_1]\ldots[\theta_n]$. On note Dom([y / M])=y et Dom([ξ / $\mathtt{K}$])=ξ.

On définit maintenant de manière formelle les notions de variable libres et liées pour les termes de TPC_{ES}. Pour cela nous allons d'abord définir les variables libres d'un terme.

$$
\begin{array}{ll}
Var(\_) & = \emptyset \\
Var(x) & = \{x\} \\
Var(\sharp z) & = \{z\} \\
Var(\langle P, Q\rangle) & = Var(P) \cup Var(Q) \\
Var((P \mid_\xi Q)) & = Var(P) \cup Var(Q) \cup \{\xi\} \\
Var(@(P, Q)) & = Var(P) \cup Var(Q)
\end{array}
$$

$$
\begin{array}{ll}
FV(x) & = \{x\} \\
FV(\mathtt{inl}_A(M)) & = FV(M) \\
FV(\mathtt{inr}_A(M)) & = FV(M) \\
FV(\langle M, N\rangle) & = FV(M) \cup FV(N) \\
FV(\lambda P.M) & = FV(M) \setminus Var(P) \\
FV(\mathtt{let}\ N\ \mathtt{be}\ P\!:\!A\ \mathtt{in}\ M) & = FV(N) \cup (FV(M) \setminus Var(P)) \\
FV(M[x/N]) & = FV(N) \cup (FV(M) \setminus \{x\}) \\
FV(M[\xi/\mathtt{K}]) & = FV(M) \setminus \{\xi\} \\
FV([M\ |_{\xi}\ N]) & = FV(M) \cup FV(N) \cup \{\xi\} \\
FV([M\ |_{\mathtt{K}}\ N]) & = FV(M) \cup FV(N) \\
FV(U\ \mathtt{of}\ N\ \mathtt{is}\ Q\!:\!A\ \mathtt{in}\ M) & = FV(U) \cup FV(N) \cup (FV(M) \setminus Var(Q))
\end{array}
$$

Les variables liées d'un terme sont définies par induction sur la structure du terme comme suit:

$$
\begin{array}{ll}
BV(x) & = \emptyset \\
BV(\mathtt{inl}_A(M)) & = BV(M) \\
BV(\mathtt{inr}_A(M)) & = BV(M) \\
BV(\langle M, N\rangle) & = BV(M) \cup BV(N) \\
BV(\lambda P.M) & = BV(M) \cup Var(P) \\
BV(\mathtt{let}\ N\ \mathtt{be}\ P\!:\!A\ \mathtt{in}\ M) & = BV(N) \cup (BV(M) \cup Var(P)) \\
BV(M[x/N]) & = BV(N) \cup (FV(M) \cup \{x\}) \\
BV(M[\xi/\mathtt{K}]) & = BV(M) \cup \{\xi\} \\
BV([M\ |_{\xi}\ N]) & = BV(M) \cup BV(N) \\
BV([M\ |_{\mathtt{K}}\ N]) & = BV(M) \cup BV(N) \\
BV(U\ \mathtt{of}\ N\ \mathtt{is}\ Q\!:\!A\ \mathtt{in}\ M) & = BV(U) \cup BV(N) \cup (BV(M) \cup Var(Q))
\end{array}
$$

Nous travaillerons toujours modulo α-conversion de telle manière que le renommage des variables liées soit automatique.

Comme dans le cas du λ-calcul, on peut définir une opération de substitution *implicite* notée $\_\{\_ \leftarrow \_\}$ qui définit le terme $M\{x \leftarrow N\}$ comme celui obtenu en remplaçant dans M toutes les occurrences libres de x par N.

Nous distinguons de plus un sous-ensemble particulier des variables libres: les variables libres de communication que l'on définit comme suit:

$$
\begin{array}{ll}
FCV(x) & = \emptyset \\
FCV(\mathtt{inl}_A(M)) & = FCV(M) \\
FCV(\mathtt{inr}_A(M)) & = FCV(M) \\
FCV(\langle M, N\rangle) & = FCV(M) \cup FCV(N) \\
FCV(\lambda P.M) & = FCV(M) \setminus Var(P) \\
FCV(\mathtt{let}\ N\ \mathtt{be}\ P\!:\!A\ \mathtt{in}\ M) & = FCV(N) \cup (FCV(M) \setminus Var(P)) \\
FCV(M[x/N]) & = FCV(N) \cup (FCV(M) \setminus \{x\}) \\
FCV(M[\xi/\mathtt{K}]) & = FCV(M) \setminus \{\xi\} \\
FCV([M\ |_{\xi}\ N]) & = FCV(M) \cup FCV(N) \cup \{\xi\} \\
FCV([M\ |_{\mathtt{K}}\ N]) & = FCV(M) \cup FCV(N) \\
FCV(z\ \mathtt{of}\ N\ \mathtt{is}\ Q\!:\!A\ \mathtt{in}\ M) & = \{z\} \cup FCV(N) \cup (FCV(M) \setminus Var(Q)) \\
FCV(\lambda P.J\ \mathtt{of}\ N\ \mathtt{is}\ Q\!:\!A\ \mathtt{in}\ M) & = FCV(\lambda P.J) \cup FCV(N) \cup (FCV(M) \setminus Var(Q))
\end{array}
$$

Définition 1.6 Un motif P est dit *linéaire* si chaque variable apparaît au plus une fois dans P.

Définition 1.7 On définit l'ensemble des *motifs acceptables* de type A, $AP(A)$, comme étant le plus petit ensemble $AP(A)$ de motifs linéaires vérifiant les conditions suivantes: $\_ \in AP(A)$; $x \in AP(A)$ pour toute variable x; $\#x \in AP(B \to C)$ pour toute variable x; $@(P,Q) \in AP(A)$ si P et $Q \in AP(A)$; $\langle P,Q\rangle \in AP(B \times C)$ et $(P \mid_\xi Q) \in AP(B+C)$ si $P \in AP(B)$ et $Q \in AP(C)$.

Les règles de typage pour TPC_{ES} sont les suivantes:

$$\frac{}{x_1{:}A_1,\ldots x_n{:}A_n \rhd x_i{:}A_i}\ (\mathrm{proj})$$

$$\frac{P{:}A,\Gamma \rhd M{:}C \qquad Q{:}B,\Gamma \rhd N{:}C}{(P|_\xi Q){:}A{+}B,\Gamma \rhd [M|_\xi N]{:}C}\ (+left)$$

$$\frac{\Gamma \rhd \lambda P{:}A.J{:}A{\to}B \qquad \Gamma \rhd M{:}A \qquad \Gamma,Q{:}B \rhd N{:}C}{\Gamma \rhd \lambda P{:}A.J\ \texttt{of}\ M\ \texttt{is}\ Q\ \texttt{in}\ N{:}C}\ (\mathrm{app})$$

$$\frac{P{:}A,Q{:}B,\Gamma \rhd M{:}C}{\langle P,Q\rangle{:}A{\times}B,\Gamma \rhd M{:}C}\ (\times left) \qquad \frac{\Gamma \rhd M{:}A \qquad \Gamma \rhd N{:}B}{\Gamma \rhd \langle M,N\rangle{:}A{\times}B}\ (\times right)$$

$$\frac{\Gamma \rhd M{:}A}{\Gamma \rhd \texttt{inl}_B(M){:}A{+}B}\ (+right_1) \qquad \frac{\Gamma \rhd M{:}B}{\Gamma \rhd \texttt{inr}_A(M){:}A{+}B}\ (+right_2)$$

$$\frac{P{:}A,\Gamma \rhd M{:}B}{\Gamma \rhd \lambda P{:}A.M{:}A{\to}B}\ (\to right) \qquad \frac{\Gamma \rhd M{:}A \qquad Q{:}B,\Gamma \rhd N{:}C}{\#z{:}A{\to}B,\Gamma \rhd z\ \texttt{of}\ M\ \texttt{is}\ Q\ \texttt{in}\ N{:}C}\ (\to left)$$

$$\frac{\Gamma \rhd M{:}A \qquad P{:}A,\Gamma \rhd N{:}B}{\Gamma \rhd \texttt{let}\ M\ \texttt{be}\ P\ \texttt{in}\ N{:}B}\ (\mathrm{let})$$

$$\frac{\Gamma \rhd M{:}A \qquad ?v{:}A,\Gamma \rhd N{:}B}{\Gamma \rhd N[v/M]{:}B}\ (\mathrm{sub})$$

$$\frac{\Gamma \rhd M{:}B}{P{:}A,\Gamma \rhd M{:}B}\ (\mathrm{Wildcard}) \qquad \frac{P_1{:}A,P_2{:}A,\Gamma \rhd M{:}B}{@(P_1,P_2){:}A,\Gamma \rhd M{:}B}\ (\mathrm{Layered})$$

$$\frac{\Gamma \rhd M{:}A}{\Gamma \rhd [M|_{\mathtt{L}} N]{:}A}\ (\mathrm{L}) \qquad \frac{\Gamma \rhd M\{\xi{\leftarrow}\mathtt{L}\}{:}A}{\Gamma \rhd M[\xi/\mathtt{L}]{:}A}\ (sub_{\mathtt{L}})$$

$$\frac{\Gamma \rhd N{:}A}{\Gamma \rhd [M|_{\mathtt{R}} N]{:}A}\ (\mathrm{R}) \qquad \frac{\Gamma \rhd M\{\xi{\leftarrow}\mathtt{R}\}{:}A}{\Gamma \rhd M[\xi/\mathtt{R}]{:}A}\ (sub_{\mathtt{R}})$$

Dans la règle (sub), $?v$ est soit une variable soit un motif flèche. Dans la règle (proj), les x_i sont deux à deux distincts. Dans la règle $(\to left)$, z est une nouvelle variable. Dans les règles $(\to left)$, (app), (let) et (sub), la condition $FCV(M) = \emptyset$ doit être vérifiée. Dans la règle (app), la condition $FCV(\lambda P.J) = \emptyset$ doit aussi être vérifiée. Dans la règle (wildcard) on doit avoir $P \in AP(A)$ et $P : A, \Gamma$ doit être linéaire.

Les règles de réduction de TPC_{ES} sont les suivantes:

$$
\begin{array}{lll}
(\lambda P.J)\ \texttt{of}\ M\ \texttt{is}\ Q\ \texttt{in}\ N & \longrightarrow_{Of} & \texttt{let}\ (\texttt{let}\ M\ \texttt{be}\ P\ \texttt{in}\ J)\ \texttt{be}\ Q\ \texttt{in}\ N \\
\texttt{let}\ \langle M_1, M_2\rangle\ \texttt{be}\ \langle P_1, P_2\rangle\ \texttt{in}\ N & \longrightarrow_{Pair} & \texttt{let}\ M_1\ \texttt{be}\ P_1\ \texttt{in}\ (\texttt{let}\ M_2\ \texttt{be}\ P_2\ \texttt{in}\ N) \\
\texttt{let}\ M\ \texttt{be}\ @(P_1, P_2)\ \texttt{in}\ N & \longrightarrow_{Cont} & \texttt{let}\ M\ \texttt{be}\ P_1\ \texttt{in}\ (\texttt{let}\ M\ \texttt{be}\ P_2\ \texttt{in}\ N) \\
\texttt{let}\ \texttt{inl}(L)\ \texttt{be}\ (P \mid_\xi Q)\ \texttt{in}\ M & \longrightarrow_{Case1} & \texttt{let}\ L\ \texttt{be}\ P\ \texttt{in}\ M[\xi/\texttt{L}] \\
\texttt{let}\ \texttt{inr}(L)\ \texttt{be}\ (P \mid_\xi Q)\ \texttt{in}\ M & \longrightarrow_{Case2} & \texttt{let}\ L\ \texttt{be}\ Q\ \texttt{in}\ M[\xi/\texttt{R}] \\
\texttt{let}\ (\lambda P.M)\ \texttt{be}\ \sharp z\ \ \texttt{in}\ N & \longrightarrow_{Lambda} & N[z/\lambda P.M] \\
\texttt{let}\ M\ \texttt{be}\ x\ \texttt{in}\ N & \longrightarrow_{Sub} & N[x/M] \\
\texttt{let}\ M\ \texttt{be}\ \_\ \texttt{in}\ N & \longrightarrow_{Weak} & N \\
[M \mid_{\texttt{L}} N] & \longrightarrow_{\texttt{L}} & M \\
[M \mid_{\texttt{R}} N] & \longrightarrow_{\texttt{R}} & N \\
x[x/M] & \longrightarrow_{Var1} & M \\
y[\theta] & \longrightarrow_{Var2} & y,\ \text{if}\ y \neq Dom(\theta) \\
(\texttt{let}\ M\ \texttt{be}\ P\ \texttt{in}\ N)[\theta] & \longrightarrow_{Dcut} & \texttt{let}\ M[\theta]\ \texttt{be}\ P\ \texttt{in}\ N[\theta] \\
\langle M, N\rangle[\theta] & \longrightarrow_{Dpair} & \langle M[\theta], N[\theta]\rangle \\
(\lambda P.M)[\theta] & \longrightarrow_{Dlambda} & \lambda P.M[\theta] \\
(z\ \texttt{of}\ M\ \texttt{is}\ P\ \texttt{in}\ N)[z/L] & \longrightarrow_{Dapp1} & L\ \texttt{of}\ M\ \texttt{is}\ P\ \texttt{in}\ N \\
(z\ \texttt{of}\ M\ \texttt{is}\ P\ \texttt{in}\ N)[\theta] & \longrightarrow_{Dapp2} & z\ \texttt{of}\ M[\theta]\ \texttt{is}\ P\ \texttt{in}\ N[\theta],\ \text{if}\ z \neq Dom(\theta) \\
(\lambda Q.J\ \texttt{of}\ M\ \texttt{is}\ P\ \texttt{in}\ N)[\theta] & \longrightarrow_{Dapp3} & \lambda Q.J[\theta]\ \texttt{of}\ M[\theta]\ \texttt{is}\ P\ \texttt{in}\ N[\theta] \\
\texttt{inl}(L)[\theta] & \longrightarrow_{Dinl} & \texttt{inl}(L[\theta]) \\
\texttt{inr}(L)[\theta] & \longrightarrow_{Dinr} & \texttt{inr}(L[\theta]) \\
[M \mid_T N][\theta] & \longrightarrow_{Dcase1} & [M[\theta] \mid_T N[\theta]],\ \text{if}\ T \neq Dom(\theta) \\
[M \mid_\xi N][\xi/\texttt{K}] & \longrightarrow_{Dcase2} & [M[\xi/\texttt{K}] \mid_{\texttt{K}} N[\xi/\texttt{K}]]
\end{array}
$$

Kesner et Cerrito ont défini dans [4] deux calculs à partir de TPC_{ES}: TPC et ES. ES est fondé sur la même grammaire que TPC_{ES}. TPC quant à lui est fondé sur la grammaire de TPC_{ES} dont on a retiré les termes de la forme M[x / M] . Nous présentons maintenant leurs règles de réduction.

$$
\begin{array}{lll}
(\lambda P.J)\ \texttt{of}\ M\ \texttt{is}\ Q\ \texttt{in}\ N & \longrightarrow_{Of} & \texttt{let}\ (\texttt{let}\ M\ \texttt{be}\ P\ \texttt{in}\ J)\ \texttt{be}\ Q\ \texttt{in}\ N \\
\texttt{let}\ \langle M_1, M_2\rangle\ \texttt{be}\ \langle P_1, P_2\rangle\ \texttt{in}\ N & \longrightarrow_{Pair} & \texttt{let}\ M_1\ \texttt{be}\ P_1\ \texttt{in}\ (\texttt{let}\ M_2\ \texttt{be}\ P_2\ \texttt{in}\ N) \\
\texttt{let}\ M\ \texttt{be}\ @(P_1, P_2)\ \texttt{in}\ N & \longrightarrow_{Cont} & \texttt{let}\ M\ \texttt{be}\ P_1\ \texttt{in}\ (\texttt{let}\ M\ \texttt{be}\ P_2\ \texttt{in}\ N) \\
\texttt{let}\ \texttt{inl}()L\ \texttt{be}\ (P \mid_\xi Q)\ \texttt{in}\ M & \longrightarrow_{Case1} & \texttt{let}\ L\ \texttt{be}\ P\ \texttt{in}\ M\{\xi \leftarrow \texttt{L}\} \\
\texttt{let}\ \texttt{inr}()L\ \texttt{be}\ (P \mid_\xi Q)\ \texttt{in}\ M & \longrightarrow_{Case2} & \texttt{let}\ L\ \texttt{be}\ Q\ \texttt{in}\ M\{\xi \leftarrow \texttt{R}\} \\
\texttt{let}\ (\lambda P.M)\ \texttt{be}\ \sharp z\ \ \texttt{in}\ N & \longrightarrow_{Lambda} & N\{z \leftarrow \lambda P.M\} \\
\texttt{let}\ M\ \texttt{be}\ x\ \texttt{in}\ N & \longrightarrow_{Sub} & N\{x \leftarrow M\} \\
\texttt{let}\ M\ \texttt{be}\ \_\ \texttt{in}\ N & \longrightarrow_{Weak} & N \\
[M \mid_{\texttt{L}} N] & \longrightarrow_{\texttt{L}} & M \\
[M \mid_{\texttt{R}} N] & \longrightarrow_{\texttt{R}} & N
\end{array}
$$

Les règles de réduction de TPC

$$
\begin{array}{lll}
x[x/M] & \longrightarrow_{Var1} & M \\
y[\theta] & \longrightarrow_{Var2} & y, \text{ if } y \neq Dom(\theta) \\
(\texttt{let } M \texttt{ be } P \texttt{ in } N)[\theta] & \longrightarrow_{Dcut} & \texttt{let } M[\theta] \texttt{ be } P \texttt{ in } N[\theta] \\
\langle M, N\rangle[\theta] & \longrightarrow_{Dpair} & \langle M[\theta], N[\theta]\rangle \\
(\lambda P.M)[\theta] & \longrightarrow_{Dlambda} & \lambda P.M[\theta] \\
(z \texttt{ of } M \texttt{ is } P \texttt{ in } N)[z/L] & \longrightarrow_{Dapp1} & L \texttt{ of } M \texttt{ is } P \texttt{ in } N \\
(z \texttt{ of } M \texttt{ is } P \texttt{ in } N)[\theta] & \longrightarrow_{Dapp2} & z \texttt{ of } M[\theta] \texttt{ is } P \texttt{ in } N[\theta], \text{ if } z \neq Dom(\theta) \\
(\lambda Q.J \texttt{ of } M \texttt{ is } P \texttt{ in } N)[\theta] & \longrightarrow_{Dapp3} & \lambda Q.J[\theta] \texttt{ of } M[\theta] \texttt{ is } P \texttt{ in } N[\theta] \\
\texttt{inl}(L)[\theta] & \longrightarrow_{Dinl} & \texttt{inl}(L[\theta]) \\
\texttt{inr}(L)[\theta] & \longrightarrow_{Dinr} & \texttt{inr}(L[\theta]) \\
[M \mid_T N][\theta] & \longrightarrow_{Dcase1} & [M[\theta] \mid_T N[\theta]], \text{ if } T \neq Dom(\theta) \\
[M \mid_\xi N][\xi/\texttt{K}] & \longrightarrow_{Dcase2} & [M[\xi/\texttt{K}] \mid_{\texttt{K}} N[\xi/\texttt{K}]]
\end{array}
$$

Les règles de réduction pour ES

TPC est la version avec substitution implicite de TPC_{ES}. ES est le sous-calcul de TPC_{ES} ne concernant que les substitutions.

Notation 1.7 Dans toute la suite on écrira M$\rightarrow_{tpces}$N pour exprimer que l'on peut appliquer une des règles de réduction de TPC_{ES} à M et que l'on obtient N. De même on écrira M$\rightarrow_{tpc}$N par rapport aux règles de réductions de TPC.

Comme dans le cas du λ-calcul avec substitution explicite [1, 2], Kesner et Cerrito ont montré [4] la propriété suivante:

Proposition 1.8

- Si M est un terme de TPC_{ES} et si M$\rightarrow_{tpces}$N alors la forme normale pour ES de M se réduit par $\rightarrow^*_{tpc}$ en la forme normale pour ES de N.
- Si M est un terme de TPC et si M$\rightarrow_{tpc}$N alors M $\rightarrow^+_{tpces}$ N

Ces deux résultats sont les lemmes (6.1) et (6.2) de [4].

Pour plus de détails sur TPC et sur TPC_{ES} on se réferrera à [4].

1.5 L'évaluation

1.5.1 Deux grand types de stratégies

Intuitivement une stratégie est un manière déterministe de conduire les calculs. Les stratégies sont vitales du point de vue de l'implantation: il faut en effet déterminiser les calculs à effectuer pour pouvoir les faire effectuer de manière automatique par une machine. Les deux classes les plus connues de stratégies sont la classe des stratégies par noms (étendue en la classe des stratégies paresseuses) et la classe des stratégies par valeurs (étendue en la classe des stratégies strictes). La première classe est une classe de stratégies qui n'effectue un calcul que quand celui-ci est nécessaire. La seconde effectue en général tous les calculs possibles dans les arguments d'une fonction avant de traiter l'appel de la fonction donnée. Cette seconde classe de stratégie bien que (souvent) non complète (certains termes ont des formes normales qui sont non atteignables par ce type de stratégies) est souvent préférée à la première pour des raisons d'encombrement mémoire. Cependant quelques langages implantent partiellement (CAML [14]) ou non (MIRANDA [16]) des stratégies par nom. Ces stratégies sont les seules capables de traiter certains problèmes sur des structures de données infinies. Nous verrons un peu plus loin deux exemples de calculs avec les stratégies du λ-calcul.

1.5.2 Evaluateurs pas à pas et d'un grand pas

Un évaluateur est un ensemble de règles de réduction déterministes, de telle sorte que pour tout terme une règle de l'évaluateur au plus s'applique. Le but premier des évaluateurs est de permettre l'implantation de stratégies de calcul en vue de définir des langages de programmation.
Il existe deux grands types d'évaluateurs, les évaluateurs d'un grand pas et les évaluateurs pas à pas. Un évaluateur d'un grand pas renvoie le résultat d'un grand pas de calcul de la stratégie qu'il implante et donne comme résultat sur un terme la "valeur" finale du terme. Un évaluateur pas à pas renvoie le résultat d'un pas de calcul pour la stratégie qu'il implante.
Chacun de ces deux types d'évaluateurs a ses avantages et ses inconvénients. Les évaluateurs d'un grand pas étant plus "globaux" comprennent souvent moins de règles de calcul et sont donc à ce titre plus faciles à appréhender. Les évaluateurs pas à pas sont plus proches de l'implantation et il arrive que certaines propriétés des stratégies qu'ils implantent soient beaucoup plus facilement démontrables.

1.5.3 Stratégies par noms et par valeurs pour le λ-calcul

La stratégie par noms La stratégie par noms pour les termes clos du λ-calcul (i.e. les termes t tels que $FV(t) = \emptyset$) est la stratégie définie par les règles de réduction de l'évaluateur pas à pas suivantes:

$$\frac{}{(\lambda x.M\ \ N) \to_N M\{x \leftarrow N\}}$$

$$\frac{M \to^N M'}{(M\ \ N) \to^N (M'\ \ N)}$$

On remarque que les réductions s'effectuent en tête d'un terme.

Nous allons maintenant donner l'évaluateur par noms paresseux d'un grand pas pour le λ-calcul. Ses règles de réduction sont les suivantes:

$$\frac{M \Downarrow^N \lambda x.L \qquad L\{x \leftarrow N\} \Downarrow^N K}{M\ \ N \Downarrow^N K}$$

$$\frac{}{\lambda x.M \Downarrow^N \lambda x.M}$$

La stratégie par valeurs La stratégie par valeurs pour les termes clos du λ-calcul est définie par les règles de calcul de l'évaluateur pas à pas suivantes:

$$\frac{}{(\lambda x.M\ \ N) \to^V M\{x \leftarrow N\}} \prec \text{Si } N \text{ est une valeur}$$

$$\frac{N \to^V N'}{(\lambda x.M\ \ N) \to^V (\lambda x.M\ \ N')} \prec \text{Si } N \text{ n' est pas une valeur}$$

$$\frac{M \to^V M'}{(M\ \ N) \to^V (M'\ \ N)}$$

où M est une valeur si et seulement si M est une λ-abstraction. Les réductions s'effectuant ici encore en tête de terme.

Nous allons maintenant donner les règles de réduction de l'évaluateur d'un grands pas pour la stratégie par valeur sur les termes clos λ-calcul.

$$\frac{M \Downarrow^V \lambda x.J \qquad N \Downarrow^V K \qquad J\{x \leftarrow K\} \Downarrow^V L}{M\ \ N \Downarrow^V L}$$

$$\frac{}{\lambda x.M \Downarrow^V \lambda x.M}$$

Le λ-calcul typé étant fortement normalisant les deux stratégies sont ici complètes. En revanche si l'on ne se restreint pas au termes typables seule la stratégie par nom est complète. Un des termes du λ-calcul classique qui montre la non complétude de la stratégie par valeurs est le terme suivant: $N = \lambda uv.v\ \Omega$ où $\Omega = (\lambda x.x\ x)\ (\lambda y.y\ y)$. En effet $N \to^N \lambda v.v\{u \leftarrow \Omega\} = \lambda v.v$ qui est une forme normale alors que $N \to^V N$ qui n'est pas une forme normale.

1.6 Notre apport

Nous avons au cours de ce stage défini huit évaluateurs d'après la subdivision suivante:

- Les évaluateurs pas à pas et les évaluateurs d'un grand pas.
- Les évaluateurs paresseux (correspondant à la stratégie par nom) et les évaluateurs stricts (correspondant à la stratégie par valeur).
- Les évaluateurs pour TPC et ceux pour TPC_{ES}.

Le tableau suivant résume rapidement les huit notations que nous utiliserons pour les huit évaluateurs que nous avons définis.

	TPC grand pas	TPC p.à.p	TPC_{ES} grand pas	TPC_{ES} p.à.p
paresseux	$\Downarrow_{l,TPC}$	$\Rightarrow_{l,TPC}$	$\Downarrow_{l,TPC_{ES}}$	$\Rightarrow_{l,TPC_{ES}}$
strict	$\Downarrow_{e,TPC}$	$\Rightarrow_{e,TPC}$	$\Downarrow_{e,TPC_{ES}}$	$\Rightarrow_{e,TPC_{ES}}$

Nous avons montré:

- Le déterminisme de ces évaluateurs.

- La *correction* de ces évaluateurs par rapport à TPC ou TPC_{ES} suivant les cas c'est à dire le fait que si M est évaluable en K par un des évaluateurs que nous avons défini alors M $\rightarrow^*_{tpc}$ K ou M $\rightarrow^*_{tpces}$ K suivant les cas. En revanche la réciproque est fausse: Si M$\rightarrow^*_{tpc}$ K avec K une "valeur" pour l'évaluateur considéré, il est possible que M ne s'évalue pas en K.

- La *complétude* de ces évaluateurs, c'est à dire que tout terme clos et bien typé de TPC est évaluable par chacun de ses évaluateurs.

- L'équivalence entre les évaluateurs pas à pas et d'un grand pas (c'est à dire le fait que l'évaluateur d'un grand pas et l'évaluateur pas à pas d'une stratégie pour un calcul calculent le même résultat).

- L'équivalence entre les évaluateurs pour TPC et ceux correspondant pour TPC_{ES}. C'est à dire que si M est un terme de TPC_{ES} qui s'évalue par un évaluateur pour TPC_{ES} en K alors la forme normale pour ES de M s'évalue par l'évaluateur correspondant pour TPC en K et réciproquement.

La section 2 donne quelques définitions générales, les sections 3 donne les quatre évaluateurs d'un grand pas et leur propriétés, la section 4 donne les quatre évaluateurs pas à pas et leur propriétés. Enfin la section 5 donne un résumé des résultats obtenus et des améliorations envisageables de ce travail.

2 Quelques généralités et définitions

Définition 2.1 On dit qu'un terme M de TPC_{ES} est *pur* si M ne contient pas de substitution explicite.

Définition 2.2 On dit qu'un terme M de TPC ou TPC_{ES} est *clos* si M ne contient pas de variable libre.

Définition 2.3 On appelle *programme* un terme clos et bien typé de TPC ou TPC_{ES}.

Propriété 2.1 On a les propriétés suivante sur ES:

- ES est confluent.

- ES termine.

- Les ES-forme normales sont des termes purs de TPC_{ES} (c'est à dire des termes de TPC).

Preuve:
On trouvera les preuves de ces propriétés dans [4]

Remarque 2.2 ES est donc fortement normalisant

Notation 2.3 Si M est un terme de TPC_{ES} on note $ES(M)$ sa forme normale pour ES.

Définition 2.4 On définit la *mesure de substitution* $\mu()$ d'un terme M de TPC_{ES} par induction sur la structure de M comme suit:

$$\begin{array}{lcl}
\mu(x) & = & 0 \\
\mu(\#z) & = & 0 \\
\mu(\mathtt{inl}(M)) & = & \mu(M) \\
\mu(\mathtt{inr}(M)) & = & \mu(M) \\
\mu(\lambda P.M) & = & \mu(M) \\
\mu(\langle M, N\rangle) & = & \mu(M) + \mu(N) \\
\mu([M \mid_{\mathtt{K}} N]) & = & \mu(M) + \mu(N) \\
\mu(\mathtt{let}\ M\ \mathtt{be}\ P\ \mathtt{in}\ N) & = & \mu(M) + \mu(N) \\
\mu(U\ \mathtt{of}\ M\ \mathtt{is}\ P\ \mathtt{in}\ N) & = & \mu(U) + \mu(M) + \mu(N) \\
\mu(M[x/N]) & = & \mu(M) + \mu(N) + 1 \\
\mu(M[\xi/\mathtt{K}]) & = & \mu(M) + 1
\end{array}$$

Intuitivement $\mu()$ compte le nombre de substitutions contenues dans un terme.

Notation 2.4 On note dans toute la suite: $\mu(\Theta)=_{def}\mu(x\Theta)$.

On sait que tout terme M peut s'écrire comme $M_1\Theta$ où M_1 n'a pas de substitution en tête.

Définition 2.5 On définit la *mesure extérieure de substitution* $\tilde{\mu}$ d'un terme $M = M_1\Theta$ par $\tilde{\mu}(M) = \tilde{\mu}(M_1\Theta) = \mu(\Theta)$

Dans toute la suite on parlera souvent d'*arbre de preuve* de l'évaluation d'un terme par un évaluateur. Ces arbres sont les traces des calculs effectués par l'évaluateur pour aboutir au résultat.

3 Les évaluateurs d'un grand pas

3.1 Un évaluateur paresseux pour TPC

L'évaluateur paresseux d'un grand pas pour TPC que l'on notera $\Downarrow_{l,TPC}$ (ou $\Downarrow_l$ quand il n'y aura pas de confusion possible), donne les règles de calculs de la stratégie paresseuse de TPC. Cette stratégie, qui correspond à la stratégie par noms du λ-calcul, est la stratégie qui n'effectue effectivement un calcul que quand celui-ci est strictement nécessaire. On peut remarquer que pour chaque terme de TPC une règle au plus de la relation $\Downarrow_{l,TPC}$ est applicable. L'évaluateur $\Downarrow_{l,TPC}$ est déterministe c'est à dire que si M est un terme de TPC tel que $M \Downarrow_{l,TPC} K^1$ et $M \Downarrow_{l,TPC} K^2$ alors $K_1 \equiv K_2$. Il possède diverses propriétés dont la correction par rapport à TPC et celle de préserver le type d'un terme typable.

Nous allons définir maintenant les formes normales paresseuses qui sont les termes de TPC résultats de l'évaluateur $\Downarrow_{l,TPC}$.

Définition 3.1 (Formes normales paresseuses) On définit les *formes normales paresseuses* (fnp) comme étant les termes clos donnés par la grammaire suivante:
K ::= $\langle M, M\rangle \mid \mathtt{inl}(M) \mid \mathtt{inr}(M) \mid \lambda P : A.M$
avec P un motif et M un terme de TPC.

Définition 3.2 (L'évaluateur $\Downarrow_{l,TPC}$) La relation $\Downarrow_{l,TPC}$ est une relation entre les termes clos de TPC et les fnp.

$$\frac{\mathtt{let}\ (\mathtt{let}\ M\ \mathtt{be}\ P\ \mathtt{in}\ J)\ \mathtt{be}\ Q\ \mathtt{in}\ N \Downarrow_{l,TPC} K}{\lambda P.J\ \mathtt{of}\ M\ \mathtt{is}\ Q\ \mathtt{in}\ N \Downarrow_{l,TPC} K}\ \text{(Of)}$$

$$\frac{M \Downarrow_{l,TPC} (M_1,M_2) \qquad \mathtt{let}\ M_1\ \mathtt{be}\ P_1\ \mathtt{in}\ (\mathtt{let}\ M_2\ \mathtt{be}\ P_2\ \mathtt{in}\ N) \Downarrow_{l,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ {<}P_1,P_2{>}\ \mathtt{in}\ N \Downarrow_{l,TPC} K}\ \text{(Prod)}$$

$$\frac{\mathtt{let}\ M\ \mathtt{be}\ P_1\ \mathtt{in}\ (\mathtt{let}\ M\ \mathtt{be}\ P_2\ \mathtt{in}\ N) \Downarrow_{l,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ @(P_1,P_2)\ \mathtt{in}\ N \Downarrow_{l,TPC} K}\ \text{(@)}$$

$$\frac{M \Downarrow_{l,TPC} \mathtt{inl}(L) \qquad \mathtt{let}\ L\ \mathtt{be}\ P\ \mathtt{in}\ N\{\xi \leftarrow \mathtt{L}\} \Downarrow_{l,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ (P|_{\xi} Q)\ \mathtt{in}\ N \Downarrow_{l,TPC} K}\ \text{(Let-left)}$$

$$\frac{M \Downarrow_{l,TPC} \mathtt{inr}(L) \qquad \mathtt{let}\ L\ \mathtt{be}\ Q\ \mathtt{in}\ N\{\xi \leftarrow \mathtt{R}\} \Downarrow_{l,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ (P|_{\xi} Q)\ \mathtt{in}\ N \Downarrow_{l,TPC} K}\ \text{(Let-right)}$$

$$\frac{M \Downarrow_{l,TPC} \lambda P.J \qquad N\{z \leftarrow \lambda P.J\} \Downarrow_{l,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ \#z\ \mathtt{in}\ N \Downarrow_{l,TPC} K}\ \text{(Let-lambda)}$$

$$\frac{N\{x \leftarrow M\} \Downarrow_{l,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ x\ \mathtt{in}\ N \Downarrow_{l,TPC} K}\ \text{(Let-var)} \qquad \frac{N \Downarrow_{l,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ \_\ \mathtt{in}\ N \Downarrow_{l,TPC} K}\ \text{(Let-weak)}$$

$$\frac{M \Downarrow_{l,TPC} K}{[M|_{\mathtt{L}} N] \Downarrow_{l,TPC} K}\ \text{(Left)} \qquad \frac{N \Downarrow_{l,TPC} K}{[M|_{\mathtt{R}} N] \Downarrow_{l,TPC} K}\ \text{(Right)}$$

$$\frac{}{{<}M,N{>} \Downarrow_{l,TPC} {<}M,N{>}}\ \text{(Ax1)} \qquad \frac{}{\lambda P.M \Downarrow_{l,TPC} \lambda P.M}\ \text{(Ax2)} \qquad \frac{}{\mathtt{inl}(M) \Downarrow_{l,TPC} \mathtt{inl}(M)}\ \text{(Ax3)}$$

$$\frac{}{\mathtt{inr}(M) \Downarrow_{l,TPC} \mathtt{inr}(M)}\ \text{(Ax4)}$$

Les règles (Prod), (Let-left), (Let-right) et (Let-lambda) sont les règles correspondant au filtrage explicite. En effet, elle expriment le fait que pour évaluer un terme de la forme `let` M `be` P `in` _ il faut d'abord évaluer M de manière à le mettre en rapport avec la forme de P.
La règle (Let-weak) exprime le fait que le motif _ est le motif servant à "oublier" le terme qu'il filtre.
Les règles (Ax_i) sont les règles "de terminaison" du calcul. Ce sont elles qui déterminent la valeur du terme évalué. Les feuilles de tout arbre de preuve de $\Downarrow_{l,TPC}$ sont forcément des règles (Ax_i).

Afin de rendre ces règles plus claires nous allons maintenant donner un exemple d'évaluation. Soit M_1 le terme de TPC suivant: $M_1 = \langle \lambda x.x, \mathtt{let}\ \lambda y.y\ \mathtt{be}\ z\ \mathtt{in}\ z\rangle$ et soit $M = \mathtt{let}\ M_1\ \mathtt{be}\ t\ \mathtt{in}\ \lambda u.t$. Nous donnons à présent l'arbre d'évaluation de M.

$$\frac{\dfrac{}{\lambda u.t\{t \leftarrow M_1\} \equiv \lambda u.M_1 \Downarrow_l \lambda u.M_1}\ \text{(Ax2)}}{M \Downarrow_l \lambda u.M_1}\ \text{(Let-var)}$$

La propriété suivante montre la correction de la définition (3.2).

Propriété 3.1 Si M est un terme clos de TPC et si $M \Downarrow_{l,TPC} K$ alors K est une fnp. Si K est une fnp alors $K \Downarrow_{l,TPC} K$.

Preuve:
La première propriété s'obtient par récurrence sur la taille de l'arbre de preuve de $M \Downarrow_l K$, la seconde est immédiate.

□

Théorème 3.2 (Déterminisme de $\Downarrow_{l,TPC}$) $\Downarrow_{l,TPC}$ est déterministe.

Preuve:
Par récurrence sur la taille l'arbre de preuve de $M \Downarrow_l K$.
Cas de base: Si $M \Downarrow_l M$ alors M $\equiv \langle N_1, N_2\rangle$, M $\equiv \mathtt{inl}(N)$, M $\equiv \mathtt{inr}(N)$ ou M $\equiv \lambda$ P. M et la propriété est clairement vérifiée.
Cas inductif: Remarquons tout d'abord à nouveau que pour tout terme pour lequel une règle s'applique cette règle est la seule à s'appliquer. Pour cela il suffit d'étudier l'ensemble des règles de l'évaluateur.

- Si M $\equiv \lambda P.J$ $\mathtt{of}$ L $\mathtt{is}$ Q $\mathtt{in}$ N, M $\equiv$ $\mathtt{let}$ M_1 $\mathtt{be}$ $@(P_1, P_2)$ $\mathtt{in}$ N, M $\equiv$ $[M \mid_{\mathtt{L}} N]$, M $\equiv [M \mid_{\mathtt{R}} N]$, M $\equiv$ $\mathtt{let}$ M $\mathtt{be}$ _ $\mathtt{in}$ N ou M $\equiv$ $\mathtt{let}$ M $\mathtt{be}$ x $\mathtt{in}$ N (c'est à dire si la règle à la racine de l'arbre a une seule prémisse) la propriété est immédiatement vérifiée par l' hypothèse de récurrence.
- Si M $\equiv$ $\mathtt{let}$ L $\mathtt{be}$ $\langle P_1, P_2\rangle$ $\mathtt{in}$ N, supposons que

  $M \Downarrow_l K^1$ provient de $L \Downarrow_l \langle\, M_1^1,\, M_2^1 \,\rangle$ et $\mathtt{let}$ M_1^1 $\mathtt{be}$ P_1 $\mathtt{in}$ $\mathtt{let}$ M_2^1 $\mathtt{be}$ P_2 $\mathtt{in}$ $N \Downarrow_l K^1$

  $M \Downarrow_l K^2$ provient de $L \Downarrow_l \langle M_1^2, M_2^2\rangle$ et $\mathtt{let}$ M_1^2 $\mathtt{be}$ P_1 $\mathtt{in}$ $\mathtt{let}$ M_2^2 $\mathtt{be}$ P_2 $\mathtt{in}$ $N \Downarrow_l K^2$

  Alors par hypothèse de récurrence on a $\langle M_1^1, M_2^1\rangle \equiv \langle M_1^2, M_2^2\rangle$ donc $\mathtt{let}$ M_1^1 $\mathtt{be}$ P_1 $\mathtt{in}$ ($\mathtt{let}$ M_2^1 $\mathtt{be}$ P_2 $\mathtt{in}$ N) $\equiv$ $\mathtt{let}$ M_1^2 $\mathtt{be}$ P_1 $\mathtt{in}$ $\mathtt{let}$ M_2^2 $\mathtt{be}$ P_2 $\mathtt{in}$ N donc par hypothèse de récurrence on a bien $K^1 \equiv K^2$. Idem si M $\equiv$ $\mathtt{let}$ M $\mathtt{be}$ $(P_1 \mid_\xi P_2)$ $\mathtt{in}$ N ou M $\equiv$ $\mathtt{let}$ M $\mathtt{be}$ $\#z$ $\mathtt{in}$ N.

□

La proposition suivante montre que $\Downarrow_{l,TPC}$ est correct par rapport à TPC.

Proposition 3.3 Si M est un terme clos de TPC et si $M \Downarrow_{l,TPC} K$ alors M $\rightarrow^*_{tpc}$ K et même M $\rightarrow^+_{tpc}$K si M n'est pas une fnp.

Preuve:
Par récurrence sur la taille de l'arbre de preuve de $M \Downarrow_l K$.
Cas de base: Si $M \Downarrow_l M$ alors $M \rightarrow^*_{tpc} M$ est évident.
Cas inductif:

- $M \equiv \lambda P.J$ $\mathtt{of}$ L $\mathtt{is}$ Q $\mathtt{in}$ $N \Downarrow_l K$ avec R=$\mathtt{let}$ ($\mathtt{let}$ L $\mathtt{be}$ P $\mathtt{in}$ J) $\mathtt{be}$ Q $\mathtt{in}$ $N \Downarrow_l K$. Par hypothèse de récurrence on a R$\rightarrow^*_{tpc}$ K. De plus on a $M \rightarrow^{Of}_{tpc} R$, donc $M \rightarrow^+_{tpc} K$.

- $M \equiv \texttt{let}\ L\ \texttt{be}\ \langle P_1, P_2\rangle\ \texttt{in}\ N \Downarrow_l K$ avec $L \Downarrow_l \langle M_1, M_2\rangle$ et R=$\texttt{let}\ M_1\ \texttt{be}\ P_1\ \texttt{in}\ (\texttt{let}\ M_2\ \texttt{be}\ P\ \texttt{in}\ N) \Downarrow_l$ K. Par hypothèse de récurrence on a L $\rightarrow^*_{tpc} \langle M_1, M_2\rangle$ donc M $\rightarrow^*_{tpc}$ $\texttt{let}\ \langle M_1, M_2\rangle\ \texttt{be}\ \langle P_1, P_2\rangle\ \texttt{in}\ N$ et par l'hypothèse de récurrence R $\rightarrow^*_{tpc}$ K, donc par application de la règle $\rightarrow^{Pair}_{tpc}$ on a bien M$\rightarrow^+_{tpc}$K.
- $M \equiv \texttt{let}\ J\ \texttt{be}\ @(P_1, P_2)\ \texttt{in}\ N \Downarrow_l K$ avec R=$\texttt{let}\ J\ \texttt{be}\ P_1\ \texttt{in}\ (\texttt{let}\ J\ \texttt{be}\ P_2\ \texttt{in}\ N) \Downarrow_l K$. Par hypothèse de récurrence on a R $\rightarrow^*_{tpc}$ K. De plus on a $M \rightarrow^{Cont}_{tpc}$ R donc M$\rightarrow^+_{tpc}$K.
- $M \equiv \texttt{let}\ J\ \texttt{be}\ (P_1 \mid_\xi P_2)\ \texttt{in}\ N \Downarrow_l K$ avec $J \Downarrow_l \texttt{inl}(L)$ et $R = \texttt{let}\ L\{\xi \leftarrow \texttt{L}\}\ \texttt{be}\ P_1\ \texttt{in}\ N \Downarrow_l K$. Par hypothèse de récurrence on a J $\rightarrow^*_{tpc}$ $\texttt{inl}(L)$ donc M $\rightarrow^*_{tpc}$ $\texttt{let}\ \texttt{inl}(L)\ \texttt{be}\ (P_1 \mid_\xi P_2)\ \texttt{in}\ N$ et $R \rightarrow^*_{tpc}$K donc par application de la règle $\rightarrow^{Case1}_{tpc}$, on a bien M$\rightarrow^+_{tpc}$ K. Le cas où $M \equiv \texttt{let}\ J\ \texttt{be}\ (P_1 \mid_\xi P_2)\ \texttt{in}\ N$ avec $J \Downarrow_l \texttt{inr}(L)$ est similaire.
- $M \equiv \texttt{let}\ J\ \texttt{be}\ \#z\ \texttt{in}\ N \Downarrow_l K$ avec $J \Downarrow_l \lambda P.L$ et $N\{z \leftarrow \lambda P.L\} \Downarrow_l K$. Par hypothèse de récurrence on a $J \rightarrow^*_{tpc} \lambda P.L$ et $N\{z \leftarrow \lambda P.L\ \} \rightarrow^*_{tpc}$ K. Donc on a M $\rightarrow^*_{tpc}$ $\texttt{let}\ \lambda P.L\ \texttt{be}\ \#z\ \texttt{in}\ N$ et, par application de la règle $\rightarrow^{Lambda}_{tpc}$, on a bien M $\rightarrow^+_{tpc}$ K.
- $M \equiv \texttt{let}\ J\ \texttt{be}\ x\ \texttt{in}\ N \Downarrow_l K$ avec $N\{x \leftarrow J\} \Downarrow_l K$. Par hypothèse de récurrence on a $N\{x \leftarrow J\} \rightarrow^*_{tpc}$ K et donc par application de la règle $\rightarrow^{Sub}_{tpc}$ on a bien M$\rightarrow^+_{tpc}$K.
- Si $M \equiv \texttt{let}\ J\ \texttt{be}\ \_\ \texttt{in}\ N \Downarrow_l K$ avec $N \Downarrow_l K$ alors par hypothèse de récurrence on a N $\rightarrow^*_{tpc}$ K et donc par application de la règle $\rightarrow^{Weak}_{tpc}$ on a bien M$\rightarrow^+_{tpc}$K.
- Les autres cas se traitent par application de la règle de TPC correspondante et par application de l'hypothèse de récurrence par une méthode similaire.

$\square$

Le théorème suivant montre que la relation $\Downarrow_{l,TPC}$ sur les termes de TPC préserve les types.

Théorème 3.4 (Préservation de types pour $\Downarrow_{l,TPC}$) Soit M un terme clos de TPC. Si $\triangleright M : A$ et $M \Downarrow_{l,TPC} K$ alors $\triangleright K : A$.

Preuve:
Ce théorème est une application immédiate de la proposition (3.3) et du théorème (5.1) de [4].

$\square$

3.2 Un évaluateur paresseux pour TPC_{ES}

L'évaluateur paresseux d'un pas pour TPC_{ES} est l'équivalent de l'évaluateur paresseux d'un pas pour TPC. Nous le noterons $\Downarrow_{l,TPC_{ES}}$ (ou $\Downarrow_l$ quand aucune confusion ne sera possible). Il se spécifie grâce à quatre relations: $\searrow$ est la relation principale. C'est cette relation qui effectue en fait le calcul. La relation $\rightsquigarrow_l$ se contente de faire "descendre" les substitutions de la tête d'un terme. Par exemple, le terme $M = (\texttt{let}\ x\ \texttt{be}\ \_\ \texttt{in}\ y)[y/M_1]$ devient par cette relation $\texttt{let}\ x[y/M_1]\ \texttt{be}\ \_\ \texttt{in}\ y[y/M_1]$. La relation $\hookrightarrow$ est un simple "purificateur". Il prend un terme de TPC_{ES} et rend sa forme purifiée. Par exemple on a: $(\texttt{let}\ x\ \texttt{be}\ \_\ \texttt{in}\ y)[y/\lambda z.z] \hookrightarrow \texttt{let}\ x\ \texttt{be}\ \_\ \texttt{in}\ \lambda z.z$. Nous avons besoin de ces deux relations, et non pas d'une seule d'entre elles comme on pourrait le supposer au premier abord, afin de renforcer le caractère paresseux de notre évaluateur. En effet $\rightsquigarrow_l$ effectue le calcul absolument indispensable pour permettre à $\searrow$ de poursuivre ses calculs tandis que $\hookrightarrow$, qui est nécessaire si l'on veut renvoyer un terme pur, purifie complètement le terme sur lequel elle est appelée. Comme $\Downarrow_{l,TPC}$, $\Downarrow_{l,TPC_{ES}}$ possède la propriété de préservation de types. Il est correct par rapport à TPC_{ES}. Enfin il est équivalent à $\Downarrow_{l,TPC}$ en ce sens que si M est un terme de TPC évaluable par $\Downarrow_{l,TPC}$ alors M est évaluable par $\Downarrow_{l,TPC_{ES}}$ et les deux évaluations donnent le même résultat et si N est un terme de TPC_{ES} évaluable par $\Downarrow_{l,TPC_{ES}}$ sa ES-forme normale est évaluable par $\Downarrow_{l,TPC}$ et les deux évaluations donnent le même résultat.

Nous allons maintenant définir les formes normales paresseuses étendues qui sont les termes de TPC_{ES} renvoyés par $\searrow$.

Définition 3.3 (Formes normales paresseuses étendues)
On définit les *formes normales paresseuses étendues* (fnp$^+$) comme étant les termes clos de TPC_{ES} donnés par la même grammaire que les fnp mais où M est un terme de TPC_{ES} (donc non obligatoirement pur).

Définition 3.4 (L'évaluateur $\Downarrow_{l,TPC_{ES}}$)
La relation $\Downarrow_{l,TPC_{ES}}$ est une relation entre les termes clos et purs de TPC_{ES} et les fnp car les programmes sont des termes clos et purs. La relation $\searrow$ est une relation entre les termes clos de TPC_{ES} et les fnp$^+$. Les relations $\rightsquigarrow_l$ et $\hookrightarrow$ sont des relations entre termes de TPC_{ES}. Comme pour $\Downarrow_{l,TPC}$, pour tout terme de TPC_{ES} une règle au plus de $\Downarrow_{l,TPC_{ES}}$ est applicable.

Dans les règles qui suivent f $\in \{\texttt{let}\ \_\ \texttt{be}\ P\ \texttt{in}\ \_; \langle \_, \_ \rangle; \lambda P.\_; \texttt{inr}(\_); \texttt{inl}(\_)\}$.

$$\frac{\texttt{let}\ (\texttt{let}\ M\ \texttt{be}\ P\ \texttt{in}\ J)\ \texttt{be}\ Q\ \texttt{in}\ N \searrow K}{\lambda P.J\ \texttt{of}\ M\ \texttt{is}\ Q\ \texttt{in}\ N \searrow K}\ (\mathrm{Of})$$

$$\frac{M \searrow \langle M_1, M_2 \rangle \qquad \texttt{let}\ M_1\ \texttt{be}\ P_1\ \texttt{in}\ (\texttt{let}\ M_2\ \texttt{be}\ P_2\ \texttt{in}\ N) \searrow K}{\texttt{let}\ M\ \texttt{be}\ {<}P_1, P_2{>}\ \texttt{in}\ N \searrow K}\ (\mathrm{Prod})$$

$$\frac{\texttt{let}\ M\ \texttt{be}\ P_1\ \texttt{in}\ (\texttt{let}\ M\ \texttt{be}\ P_2\ \texttt{in}\ N) \searrow K}{\texttt{let}\ M\ \texttt{be}\ @(P_1, P_2)\ \texttt{in}\ N \searrow K}\ (@)$$

$$\frac{M \searrow \texttt{inl}(L) \qquad \texttt{let}\ L\ \texttt{be}\ P\ \texttt{in}\ N[\xi/\mathtt{L}] \searrow K}{\texttt{let}\ M\ \texttt{be}\ (P|_{\xi} Q)\ \texttt{in}\ N \searrow K}\ (\text{Let-left})$$

$$\frac{M \searrow \texttt{inr}(L) \qquad \texttt{let}\ L\ \texttt{be}\ Q\ \texttt{in}\ N[\xi/\mathtt{R}] \searrow K}{\texttt{let}\ M\ \texttt{be}\ (P|_{\xi} Q)\ \texttt{in}\ N \searrow K}\ (\text{Let-right})$$

$$\frac{M \searrow \lambda P.J \qquad N[z/\lambda P.J] \searrow K}{\texttt{let}\ M\ \texttt{be}\ \#z\ \texttt{in}\ N \searrow K}\ (\text{Let-lambda})$$

$$\frac{N[x/M] \searrow K}{\texttt{let}\ M\ \texttt{be}\ x\ \texttt{in}\ N \searrow K}\ (\text{Let-var}) \qquad \frac{N \searrow K}{\texttt{let}\ M\ \texttt{be}\ \_\ \texttt{in}\ N \searrow K}\ (\text{Let-weak})$$

$$\frac{M\searrow K}{[M|_{\mathtt{L}}N]\searrow K}\ (\text{Left}) \qquad \frac{N\searrow K}{[M|_{\mathtt{R}}N]\searrow K}\ (\text{Right})$$

$$\frac{M[\theta]\rightsquigarrow_l M_1 \qquad M_1\Theta\searrow K}{M[\theta]\Theta\searrow K}\ (\text{Down})$$

$$\frac{}{<M,N>\searrow<M,N>}\ (\text{Ax1}) \qquad \frac{}{\lambda P.M\searrow\lambda P.M}\ (\text{Ax2}) \qquad \frac{}{\texttt{inl}(M)\searrow\texttt{inl}(M)}\ (\text{Ax3})$$

$$\frac{}{\texttt{inr}(M)\searrow\texttt{inr}(M)}\ (\text{Ax4})$$

$$\frac{}{x[x/M]\rightsquigarrow_l M} \qquad \frac{}{y[\theta]\rightsquigarrow_l y}\prec \text{si } y\neq Dom(\theta)$$

$$\frac{}{f(M_1,\cdots,M_n)[\theta]\rightsquigarrow_l f(M_1[\theta],\cdots,M_n[\theta])}$$

$$\frac{}{(z\ \texttt{of}\ M\ \texttt{is}\ P\ \texttt{in}\ N)[z/L]\rightsquigarrow_l L\ \texttt{of}\ M\ \texttt{is}\ P\ \texttt{in}\ N}$$

$$\frac{}{(z\ \texttt{of}\ M\ \texttt{is}\ P\ \texttt{in}\ N)[\theta]\rightsquigarrow_l z\ \texttt{of}\ M[\theta]\ \texttt{is}\ P\ \texttt{in}\ N[\theta]}\prec \text{si } z\neq Dom(\theta)$$

$$\frac{}{(\lambda Q.J\ \texttt{of}\ M\ \texttt{is}\ P\ \texttt{in}\ N)[\theta]\rightsquigarrow_l \lambda Q.J[\theta]\ \texttt{of}\ M[\theta]\ \texttt{is}\ P\ \texttt{in}\ N[\theta]}$$

$$\frac{}{[M|_{\mathtt{T}}N][\theta]\rightsquigarrow_l [M[\theta]|_{\mathtt{T}}N[\theta]]}\prec \text{si } \mathtt{T}\neq Dom(\theta)$$

$$\frac{}{[M|_{\xi}N][\xi/\mathtt{K}]\rightsquigarrow_l [M[\xi/\mathtt{K}]|_{\mathtt{K}}N[\xi/\mathtt{K}]]}$$

$$\frac{}{M\hookrightarrow M}\prec \text{si M est pur}$$

$$\frac{M\Theta\hookrightarrow M_1}{x[x/M]\Theta\hookrightarrow M_1} \qquad \frac{y\Theta\hookrightarrow M_1}{y[\theta]\Theta\hookrightarrow M_1}\prec \text{si } y\neq Dom(\theta)$$

$$\frac{M_1\Theta\hookrightarrow K_1\cdots M_n\Theta\hookrightarrow K_n}{f(M_1,\cdots,M_N)\Theta\hookrightarrow f(K_1,\cdots,K_N)}\prec \text{si } \Theta \text{ n'est pas vide ou si l'un des } M_i \text{ n'est pas pur}$$

$$\frac{(L\ \texttt{of}\ M\ \texttt{is}\ P\ \texttt{in}\ N)\Theta\hookrightarrow M_1}{(z\ \texttt{of}\ M\ \texttt{is}\ P\ \texttt{in}\ N)[z/L]\Theta\hookrightarrow M_1}$$

$$\frac{(z\ \texttt{of}\ M[\theta]\ \texttt{is}\ P\ \texttt{in}\ N[\theta])\Theta\hookrightarrow K}{(z\ \texttt{of}\ M\ \texttt{is}\ P\ \texttt{in}\ N)[\theta]\Theta\hookrightarrow K}\prec \text{si } z\neq Dom(\theta)$$

$$\frac{J\Theta\hookrightarrow J_1 \qquad M\Theta\hookrightarrow M_1 \qquad N\Theta\hookrightarrow N_1}{(\lambda P.J\ \texttt{of}\ M\ \texttt{is}\ Q\ \texttt{in}\ N)\Theta\hookrightarrow\lambda P.J_1\ \texttt{of}\ M_1\ \texttt{is}\ Q\ \texttt{in}\ N_1}\prec \text{Si } \Theta \text{ n'est pas vide ou } J,\ M \text{ ou } N \text{ n'est pas pur}$$

$$\frac{[M[\theta]|_{\mathtt{T}}N[\theta]]\Theta\hookrightarrow K}{[M|_{\mathtt{T}}N][\theta]\Theta\hookrightarrow K}\prec \text{si } T\neq Dom(\theta)$$

$$\frac{M[\xi/\mathtt{K}]\Theta\hookrightarrow M_1 \qquad N[\xi/\mathtt{K}]\Theta\hookrightarrow N_1}{[M|_{\xi}N][\xi/\mathtt{K}]\Theta\hookrightarrow[M_1|_{\mathtt{K}}N_1]}$$

$$\frac{M\hookrightarrow M_1 \qquad N\hookrightarrow N_1}{[M|_{\mathtt{T}}N]\hookrightarrow[M_1|_{\mathtt{T}}N_1]}\prec \text{si M ou N n'est pas pur}$$

$$\frac{M\searrow L \qquad L\hookrightarrow K}{M\Downarrow_{l,TPC_{ES}}K}$$

Comme pour l'évaluateur $\Downarrow_{l,TPC}$, les règles (Prod), (Let-left), (Let-right) et (Let-lambda) sont les règles correspondant au filtrage explicite.

La règle (Let-weak) exprime le fait que le motif _ est le motif servant à "oublier" le terme qu'il filtre.
Les règles (Ax_i) sont les règles "de terminaison" du calcul. Ce sont elles qui déterminent la valeur du termes évalué. Les feuilles de tout arbre de preuve de $\Downarrow_{l,TPC}$ sont forcément des règles (Ax_i) ou des règles $\dfrac{}{M \hookrightarrow M}$ avec M pur.
La règle (Down) est la règle qui permet de faire appel à la relation $\rightsquigarrow_l$ et ainsi de faire descendre les substitutions de la tête du terme que l'on est entrain d'évaluer.
La règle (Eval) est la règle qui exprime le fait que pour évaluer un terme, on commence par effectuer le calcul puis on purifie le "pré-résultat" obtenu.

Comme pour la section précédente, nous donnons maintenant un exemple d'évaluation. Soit M_1 le terme de TPC_{ES} suivant: $M_1 = \langle \lambda x.x, \texttt{let } \lambda y.y \texttt{ be } z \texttt{ in } z \rangle$ et soit $M = \texttt{let } M_1 \texttt{ be } t \texttt{ in } \lambda u.t$. Nous donnons à présent l'arbre d'évaluation de M.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{(\lambda u.t)[t/M_1] \rightsquigarrow_l \lambda u.t[t/M_1] \equiv M_2} \qquad \dfrac{}{M_2 \searrow M_2}\,(\mathrm{Ax2})}{(\lambda u.t)[t/M_1] \searrow M_2}\,(\mathrm{Down})}{M \searrow M_2}\,(\mathrm{Let\text{-}var}) \qquad \dfrac{\dfrac{\dfrac{}{M_1 \hookrightarrow M_1}}{t[t/M_1] \hookrightarrow M_1}}{M_2 \hookrightarrow \lambda u.M_1}\,(\mathrm{fct})}{M \Downarrow_l \lambda u.M_1}\,(\mathrm{Eval})$$

Dans la suite de cette section on fera régulièrement des preuves par récurrence sur la taille d'arbre de preuve de $\Downarrow_{l,TPC_{ES}}$. Nous remarquons dès maintenant que le cas de base d'une telle récurrence est un cas de la forme suivante:
$$\dfrac{M \searrow M \qquad M \hookrightarrow M}{M \Downarrow_{l,TPC_{ES}} M}$$

Propriété 3.5

1. $M \hookrightarrow K \Leftrightarrow$ K=$ES(M)$.
2. Si M $\equiv g(M_1, \ldots, M_n)\Theta$ avec g un constructeur autre que la substitution et si $\forall i, M_i\Theta \hookrightarrow K_i$ alors $M \hookrightarrow g(K_i, \ldots, K_n)$.
3. Si $M \searrow K$ alors K est une fnp$^+$ et si K est une fnp$^+$ alors $K \searrow K$.

<u>Preuve:</u>
La propriété 1 s'obtient par une simple récurrence sur la taille de l'arbre de preuve de $M \hookrightarrow K$. La propriété 2 s'obtient par une simple récurrence sur la somme des tailles des arbres de preuve de $M_i\Theta \hookrightarrow K_i$. La propriété 3 s'obtient par une récurrence sur la taille de l'arbre de preuve de $M \searrow K$.

Remarque 3.6

1. L'étude de l'évaluateur $\hookrightarrow$ montre que quelque soit le terme M de TPC_{ES} et quelque soit Θ il existe M_1 tel que $M\Theta \hookrightarrow M_1$.

2. Si $M \hookrightarrow N \equiv g(K_1, \ldots, K_n)$ avec g n'importe quel autre constructeur que la substitution deux cas sont possibles: soit M $\equiv g(M_1, \ldots, M_n)\Theta$, avec Θ éventuellement vide, soit $M \equiv x[\theta]\Theta$.

3. Si $M \searrow L$ alors il existe K tel que $M \Downarrow_{l,TPC_{ES}} K$ (il suffit de prendre K=$ES(L)$ par la propriété (3.5-1)).

Lemme 3.7 Si M est un terme clos de TPC_{ES} et si $M \Downarrow_{l,TPC_{ES}} K$ alors K est une fnp. Si K est une fnp alors $K \Downarrow_{l,TPC_{ES}} K$.

Preuve:
Si $M \Downarrow_l K$ alors par définition de $\Downarrow_l$ il existe L tel que $M \searrow L$ et $L \hookrightarrow K$. Par la propriété (3.5-3), L est une fnp$^+$. Par la propriété (3.5-1), $K = ES(L)$ donc K est pur et K est une fnp$^+$ donc K est bien une fnp.

□

Notation 3.8 Si M est un terme clos de TPC_{ES} évaluable en K par $\Downarrow_{l,TPC_{ES}}$, on défini $\nu(M, K)$ comme étant la taille de l'arbre de preuve de M $\Downarrow_{l,TPC_{ES}}$ K.

Proposition 3.9 Si N $\equiv N_1[\theta]\Theta$ et $N_1[\theta] \rightsquigarrow_l N_2$, alors $\tilde{\mu}(N_2\Theta) < \tilde{\mu}(N_1[\theta]\Theta) = \tilde{\mu}(N)$

Preuve:
Par récurrence sur la taille de $[\theta]\Theta$ par définition de $\tilde{\mu}(N)$.

□

La propriété suivante montre que deux termes clos de TPC_{ES} ayant la même forme pure (la même ES-forme normale) s'évaluent par $\Downarrow_{l,TPC_{ES}}$en le même terme.

Proposition 3.10 (Cohérence de $\Downarrow_{l,TPC_{ES}}$) Si M et N sont deux termes clos et si $ES(M)$=$ES(N)$ et $M \Downarrow_{l,TPC_{ES}} K$ alors $N \Downarrow_{l,TPC_{ES}} K$.

Preuve:
Par induction sur $\prec$ l'ordre lexicographique ($\nu(M,K)$, $\tilde{\mu}(N)$).
Cas de base: Si M $\Downarrow_l$ M et $\tilde{\mu}(N)$=0 alors M est pur et donc ES(M)=M=ES(N). Puisque $\tilde{\mu}(N)$=0, on sait que M et N ont le même constructeur de tête par la remarque (3.6-2). Par le lemme (3.7), on sait que M est une fnp. Donc N est une fnp$^+$. Par la propriété (3.5-3), N $\searrow$ N. Par la propriété (3.5-1) on a N $\hookrightarrow$ M. Donc on a N $\Downarrow_l$ M.
Cas inductif: On a deux cas possibles:

1. $\tilde{\mu}(N)$>0 alors il existe nécessairement N_1 sans substitution en tête et N_2 tels que $N = N_1[\theta]\Theta$ et $N_1[\theta] \rightsquigarrow_l N_2$. Par la proposition (3.9), $\tilde{\mu}(N_2\Theta)$<$\tilde{\mu}(N)$. Donc $((M\ , K), N_2\Theta) \prec ((M\ , K), N)$. Par hypothèse d'induction on a $N_2\Theta \Downarrow_l K$. On a alors: $$\dfrac{\dfrac{N_1[\theta] \rightsquigarrow_l N_2 \qquad N_2\Theta \searrow L}{N \searrow L} \qquad L \hookrightarrow K}{N \Downarrow_l K}$$
Donc on a, $N \Downarrow_l K$.

2. Si $\tilde{\mu}(N)$=0 alors

   - M $\equiv \lambda P.J_1$ `of` M_1 `is` Q `in` N_1 avec R_1 =`let (let` M_1 `be` P `in` M_1) `be` Q `in` $N_1 \Downarrow_l K$. Grâce à l'hypothèse $\tilde{\mu}(N)$=0, on a: N $\equiv \lambda P.J_2$ `of` M_2 `is` Q `in` N_2. On pose alors R_2 =`let (let` M_2 `be` P `in` J_2) `be` Q `in` N_2. Il est évident que l'hypothèse ES(M) = ES(N) implique ES(R_1) = ES(R_2). De plus on a clairement $((R_1,K),R_2) \prec((M,K),\mathrm{N})$. Donc par hypothèse d'induction $R_2 \Downarrow_l K$ d'où $N \Downarrow_l K$.
   - Le cas où $M \equiv$ `let` M_1 `be` $@(P_1,P_2)$ `in` N, $M \equiv$ `let` M_1 `be` x `in` N, $M \equiv$ `let` M `be _ in` N, $M \equiv [M_1 |_{\mathtt{L}} N]$ ou $M \equiv [M_1 |_{\mathtt{R}} N]$ se traitent de manière similaire.
   - $M \equiv$ `let` M_1 `be` $\langle P_1,P_2\rangle$ `in` $N_1 \Downarrow_l K$ avec $M_1 \searrow \langle J_1^1, J_2^1\rangle$ et R_1 =`let` J_1^1 `be` P_1 `in (let` J_2^1 `be` P_2 `in` N_1) $\Downarrow_l K$. Par la remarque (3.6-3) il existe $\langle J_1,J_2\rangle$ tel que $M_1 \Downarrow_l \langle J_1,J_2\rangle$ avec $J_1 = ES(J_1^1)$ et $J_2 = ES(J_2^1)$. Puisque $\tilde{\mu}(N)$=0, N $\equiv$ `let` M_2 `be` $\langle P_1,P_2\rangle$ `in` N_2. Grâce a l'hypothèse $ES(M)$=$ES(N)$, on a $ES(M_2)$=$ES(M_1)$ et $ES(N_2)$=$ES(N_1)$. Par l'hypothèse d'induction, puisque $((M_1, \langle J_1, J_2 \rangle), M_2) \prec((M,K),N)$, on a $M_2 \Downarrow_l \langle J_1,J_2\rangle$ donc par définition, il existe $\langle J_1^2,J_2^2\rangle$ tel que $M_2 \searrow \langle J_1^2,J_2^2\rangle$ avec $ES(J_1^2) = J_1$ et $ES(J_2^2)$=J_2. On a alors par l'hypothèse d'induction R_2=`let` J_1^2 `be` P_1 `in (let` J_2^2 `be` P_2 `in` N_2) $\Downarrow_l K$ donc par définition il existe L tel que $R_2 \searrow L$ et $L \hookrightarrow K$. On peut donc conclure $N \Downarrow_l K$.
   - Les cas où $M \equiv$ `let` M_1 `be` $(P |_\xi Q)$ `in` N ou $M \equiv$ `let` M `be` $\#z$ `in` N se traitent de manière similaire.
   - M est une fnp$^+$. $\tilde{\mu}(N)$=0 donc N est une fnp$^+$ et par la propriété (3.5-1) on a $N \hookrightarrow K$. On peut alors conclure que $N \Downarrow_l K$.
   - $M \equiv M_1[\theta]\Theta \Downarrow_l K$ avec $M_1[\theta] \rightsquigarrow_l M_2$, $M_2\Theta \Downarrow_l K$ et M_1 n'a pas de substitution en tête. On a clairement $((M_2\Theta, \langle J_1^2, J_2^2 \rangle),N) \prec((M,K),N)$ donc par hypothèse d'induction on a bien N $\Downarrow_l$ K.

□

On en déduit en particulier la proposition suivante:

Proposition 3.11 Si M est un terme clos de TPC_{ES} alors: $M \Downarrow_{l,TPC_{ES}} K \Leftrightarrow ES(M) \Downarrow_{l,TPC_{ES}} K$

Preuve:
C'est une application immédiate de la proposition (3.10) puisque $ES(ES(M)) = ES(M)$.

□

Nous allons maintenant montrer l'équivalence annoncée entre $\Downarrow_{l,TPC_{ES}}$ et $\Downarrow_{l,TPC_{ES}}$. Cette équivalence signifie que la stratégie paresseuse avec substitutions explicite implante bien la stratégie paresseuse avec substitution implicite.

Théorème 3.12 (Equivalence entre $\Downarrow_{l,TPC_{ES}}$ et $\Downarrow_{l,TPC}$ (I)) Si M est un terme clos de TPC_{ES} et si $M \Downarrow_{l,TPC_{ES}} K$ alors $ES(M) \Downarrow_{l,TPC} K$.

Preuve:
Par récurrence sur la taille de l'arbre de preuve de $M \Downarrow_{l,TPC_{ES}} K$.
Cas de base: Si $M \Downarrow_{l,TPC_{ES}} M$ alors, par le lemme (3.7), M est une fnp donc $ES(M){=}M$ et la propriété est vérifiée.
Cas inductif:

- $M \equiv \lambda P.J$ **of** M_1 **is** Q **in** $N \Downarrow_{l,TPC_{ES}} K$ alors le résultat est clairement vrai par hypothèse de récurrence.

- $M \equiv$ `let` M_1 `be` $\langle P_1, P_2\rangle$ `in` $N \Downarrow_{l,TPC_{ES}} K$ avec $M_1 \searrow \langle J_1, J_2\rangle$ et `let` J_1 `be` P_1 `in` (`let` J_2 `be` P_2 `in` $N) \Downarrow_{l,TPC_{ES}} K$. Par hypothèse de récurrence et par la remarque (3.6-1) il existe $\langle L_1, L_2\rangle$ tel que $ES(M_1) \Downarrow_{l,TPC} \langle L_1, L_2\rangle$ avec $L_1{=}ES(J_1)$ et $L_2{=}ES(J_2)$. Par hypothèse de récurrence `let` L_1 `be` P_1 `in` (`let` L_2 `be` P_2 `in` ES(N)) $\Downarrow_{l,TPC} K$ donc on a bien $ES(M) \Downarrow_{l,TPC} K$.

- Les autres cas sauf celui où M est une fnp$^+$ et celui où M a une substitution en tête se traitent par une méthode similaire à un des deux précédents.

- $M \equiv M_1[\theta]\Theta \Downarrow_{l,TPC_{ES}} K$. On a alors: $\dfrac{\dfrac{M_1[\theta]\rightsquigarrow_l M_2 \qquad M_2\Theta\searrow L}{M_1[\theta]\Theta\searrow L} \qquad L\hookrightarrow K}{M_1[\theta]\Theta\Downarrow_{l,TPC_{ES}}K}$.

  Donc $M_2\Theta \Downarrow_l K$ comme le montre l'arbre de preuve suivant: $\dfrac{M_2\Theta\searrow L \qquad L\hookrightarrow K}{M_2\Theta\Downarrow_{l,TPC_{ES}}K}$
  . De plus il est clair que le second arbre est de taille strictement inférieure à celle du premier donc par hypothèse de récurrence $ES(M_2\Theta) \Downarrow_{l,TPC} K$ mais $ES(M_2\Theta){=}ES(M)$ d'où le résultat.

- M est une fnp$^+$ alors $ES(M)$ est une fnp et $M \Downarrow_{l,TPC_{ES}} ES(M)$ d'où le résultat.

$\square$

Théorème 3.13 (Equivalence entre $\Downarrow_{l,TPC_{ES}}$ et $\Downarrow_{l,TPC}$ (II)) Si M est un terme clos de TPC et si $M \Downarrow_{l,TPC} K$ alors $M \Downarrow_{l,TPC_{ES}} K$.

Preuve:
Par récurrence sur la taille de l'arbre de preuve de $M \Downarrow_{l,TPC} K$.
Cas de base: Si $M \Downarrow_{l,TPC} M$ alors M est une fnp et par le lemme (3.7) on a bien $M \Downarrow_{l,TPC_{ES}} M$.
Cas inductif:

- Si $M \equiv \lambda P.J$ **of** M_1 **is** Q **in** $N \Downarrow_{l,TPC} K$ avec `let` (`let` M_1 `be` P `in` J) `be` Q `in` $N \Downarrow_{l,TPC} K$ alors par hypothèse de récurrence `let` (`let` M_1 `be` P `in` J) `be` Q `in` $N \Downarrow_{l,TPC_{ES}} K$. Il existe donc L tel que `let` (`let` M_1 `be` P `in` J) `be` Q `in` $N \searrow L$ et $L \hookrightarrow K$. On peut donc conclure que $M \Downarrow_{l,TPC_{ES}} K$.

- Si $M \equiv$ `let` J `be` $\langle P_1, P_2\rangle$ `in` $N \Downarrow_{l,TPC} K$ avec $J \Downarrow_{l,TPC} \langle M_1, M_2\rangle$ et `let` M_1 `be` P_1 `in` (`let` M_2 `be` P_2 `in` $N) \Downarrow_{l,TPC} K$ alors par hypothèse de récurrence $J \Downarrow_{l,TPC_{ES}} \langle M_1, M_2\rangle$. Donc il existe $\langle M_1', M_2'\rangle$ tel que $J \searrow \langle M_1', M_2'\rangle$ et $\langle M_1', M_2'\rangle \hookrightarrow \langle M_1, M_2\rangle$. Par hypothèse de récurrence `let`

M_1 `be` P_1 `in` (`let` M_2 `be` P_2 `in` N) $\Downarrow_{l,TPC_{ES}} K$. De plus comme $ES(\langle M_1, M_2\rangle) = ES(\langle M_1, M_2\rangle)$, par la proposition (3.10) `let` M_1' `be` P_1 `in` (`let` M_2' `be` P_2 `in` N) $\Downarrow_{l,TPC_{ES}} K$. Donc on a bien $M \Downarrow_{l,TPC_{ES}} K$.

- Tous les cas ne faisant pas apparaître de substitution se traitent de la même manière que les deux cas précédents.
- Si $M \equiv$ `let` M_1 `be` x `in` $N \Downarrow_{l,TPC} K$ avec $N\{x \leftarrow M_1\} \Downarrow_{l,TPC} K$ alors par hypothèse de récurrence $N\{x \leftarrow M_1\} \Downarrow_{l,TPC_{ES}} K$, par la proposition (3.10), puisque $ES(N[x/M_1])=ES(N\{x \leftarrow M_1\})$, $N[x/M_1] \Downarrow_{l,TPC_{ES}} K$. Donc on a bien $M \Downarrow_{l,TPC_{ES}} K$.
- Tous les autres cas se traitent d'une manière similaire au cas précédent.

□

On a par application immédiate de ce théorème et du théorème (3.14) pour l'évaluateur paresseux de TPC le résultat suivant.

Théorème 3.14 (Déterminisme de $\Downarrow_{l,TPC_{ES}}$) $\Downarrow_{l,TPC_{ES}}$ est déterministe.

Proposition 3.15 Si M est un terme clos de TPC_{ES} et si $M \Downarrow_{l,TPC_{ES}} K$ alors M $\rightarrow^*_{tpces}$ K et même M $\rightarrow^+_{tpces}$ K si M n'est pas une fnp.

<u>Preuve:</u>
On a par le théorème (3.12), ES(M) $\Downarrow_{l,TPC}$ K. Par la proposition (3.3) on a donc ES(M) $\rightarrow^*_{tpc} K$. On a clairement $M \rightarrow^*_{tpces}$ ES(M). De plus par le lemme (6.3) de [4] on a ES(M)$\rightarrow^*_{tpces} K$. Donc finalement on a $M \rightarrow^*_{tpces} K$. De plus si M n'est pas une fnp alors deux cas sont possibles: soit ES(M) $\rightarrow^+_{tpc} K$ soit $M \rightarrow^+_{tpces} ES(M)$. Dans les deux cas on a bien $M \rightarrow^+_{tpces} K$.

□

Théorème 3.16 (Préservation de types pour $\Downarrow_{l,TPC_{ES}}$) Si M est un terme clos de TPC_{ES} et si $M \Downarrow_{l,TPC_{ES}} K$ et si $\triangleright M : A$ alors $\triangleright K : A$

<u>Preuve:</u>
Ce théorème est une application immédiate de la propriété (3.15) et du théorème (3.8) de [4].

□

3.3 Un évaluateur strict pour TPC

L'évaluateur strict d'un grand pas pour TPC est un évaluateur correspondant à la stratégie par valeurs du λ calcul. Nous le noterons $\Downarrow_{e,TPC}$ (ou $\Downarrow_e$ quand aucune confusion ne sera possible). Comme les évaluateurs précédents $\Downarrow_e$ possèdes les propriétés d'être déterministe, correct par rapport à TPC et de préserver le type des termes.

On définit ici les formes normales strictes qui sont les termes de TPC résultats de $\Downarrow_{e,TPC}$.

Définition 3.5 (Formes normales strictes) On appelle *formes normales strictes* (fns) les termes clos donnés par la grammaire suivante:
K : : $= \langle K, K\rangle \mid \mathtt{inl}(K) \mid \mathtt{inr}(K) \mid \lambda P.M$
où P est un motif et M un terme clos de TPC.

Cette définition diffère fondamentalement de la définition (3.1). En effet, dans le cas de fns, on évalue à l'intérieur constructeurs $\langle\ \_, \_\ \rangle$, $\mathtt{inl}(\_)$ et $\mathtt{inr}(\_)$ ce qui n'était pas le cas de fnp.

Définition 3.6 (L'évaluateur $\Downarrow_{e,TPC}$) La relation $\Downarrow_{e,TPC}$ est une relation entre les termes clos de TPC et les fns.

$$\frac{\mathtt{let}\ (\mathtt{let}\ M\ \mathtt{be}\ P\ \mathtt{in}\ J)\ \mathtt{be}\ Q\ \mathtt{in}\ N \Downarrow_{e,TPC} K}{\lambda P.J\ \mathtt{of}\ M\ \mathtt{is}\ Q\ \mathtt{in}\ N \Downarrow_{e,TPC} K}\ (\mathrm{Of})$$

$$\frac{M \Downarrow_{e,TPC} \langle M_1, M_2\rangle \qquad \mathtt{let}\ M_1\ \mathtt{be}\ P_1\ \mathtt{in}\ (\mathtt{let}\ M_2\ \mathtt{be}\ P_2\ \mathtt{in}\ N) \Downarrow_{e,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ \langle P_1, P_2\rangle\ \mathtt{in}\ N \Downarrow_{e,TPC} K}\ (\mathrm{Prod})$$

$$\frac{\mathtt{let}\ M\ \mathtt{be}\ P_1\ \mathtt{in}\ \mathtt{let}\ M_1\ \mathtt{be}\ P_2\ \mathtt{in}\ N \Downarrow_{e,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ @(P_1, P_2)\ \mathtt{in}\ N \Downarrow_{e,TPC} K}\ (@)$$

$$\frac{M \Downarrow_{e,TPC} \mathtt{inl}(L) \qquad \mathtt{let}\ L\ \mathtt{be}\ P\ \mathtt{in}\ N\{\xi \leftarrow \mathtt{L}\} \Downarrow_{e,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ (P|_{\xi} Q)\ \mathtt{in}\ N \Downarrow_{e,TPC} K}\ (\text{Let-left})$$

$$\frac{M \Downarrow_{e,TPC} \mathtt{inr}(L) \qquad \mathtt{let}\ L\ \mathtt{be}\ Q\ \mathtt{in}\ N\{\xi \leftarrow \mathtt{R}\} \Downarrow_{e,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ (P|_{\xi} Q)\ \mathtt{in}\ N \Downarrow_{e,TPC} K}\ (\text{Let-right})$$

$$\frac{M \Downarrow_{e,TPC} \lambda P.J \qquad N\{z \leftarrow \lambda P.J\} \Downarrow_{e,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ \#z\ \mathtt{in}\ N \Downarrow_{e,TPC} K}\ (\text{Let-lambda})$$

$$\frac{M \Downarrow_{e,TPC} J \qquad N\{x \leftarrow J\} \Downarrow_{e,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ x\ \mathtt{in}\ N \Downarrow_{e,TPC} K}\ (\text{Let-var}) \qquad \frac{N \Downarrow_{e,TPC} K}{\mathtt{let}\ M\ \mathtt{be}\ \_\ \mathtt{in}\ N \Downarrow_{e,TPC} K}\ (\text{Let-weak})$$

$$\frac{M \Downarrow_{e,TPC} K}{[M|_{\mathtt{L}} N] \Downarrow_{e,TPC} K}\ (\mathrm{Left}) \qquad \frac{N \Downarrow_{e,TPC} K}{[M|_{\mathtt{R}} N] \Downarrow_{e,TPC} K}\ (\mathrm{Right})$$

$$\frac{M \Downarrow_{e,TPC} L}{\mathtt{inl}(M) \Downarrow_{e,TPC} \mathtt{inl}(L)}\ (\mathrm{Inl}) \qquad \frac{M \Downarrow_{e,TPC} L}{\mathtt{inr}(M) \Downarrow_{e,TPC} \mathtt{inr}(L)}\ (\mathrm{Inr})$$

$$\frac{M \Downarrow_{e,TPC} K \qquad N \Downarrow_{e,TPC} L}{\langle M, N\rangle \Downarrow_{e,TPC} \langle K, L\rangle}\ (\mathrm{Pair})$$

$$\frac{}{\lambda P.M \Downarrow_{e,TPC} \lambda P.M}\ (\mathrm{Ax})$$

Cet évaluateur montre l'un des progrès de TPC et TPC_{ES} par rapport au calcul de Tannen, Kesner et Puel [13]. En effet, dans ce dernier calcul, on évaluait dans la stratégie stricte le sous-terme M dans le $\mathtt{let}\ M\ \mathtt{be}\ \_\ \mathtt{in}\ N$ ce qui était clairement non nécessaire alors que dans TPC (et cela sera encore vrai pour TPC_{ES}) cette évaluation n'a pas lieu d' être faite, car le filtrage explicite rend inutile ce type d'évaluation.
Comme dans le cas des deux évaluateurs précédents, les règles (Prod), (Let-left), (Let-right) et (Let-lambda) sont les règles du filtrage. D'une manière générales l'ensemble de règles de réduction de $\Downarrow_{e,TPC}$ est assez semblable à celui de $\Downarrow_{l,TPC}$. La principale différence est le remplacement des règles (Ax_1), (Ax_3) et (Ax_4) par les règles (Pair), (Inl) et (Inr). C'est de ce remplacement que vient le caractère strict de cet évaluateur. La règle (Let-var) change aussi par rapport

à la définition de l'évaluateur paresseux afin d'obliger le terme M à être évalué en premier.

De nouveau nous donnons un exemple d'évaluation.soit M_1 le terme de TPC suivant: $M_1 = \langle \lambda x.x, \texttt{let}\ \lambda y.y\ \texttt{be}\ z\ \texttt{in}\ z \rangle$, alors $M = \texttt{let}\ M_1\ \texttt{be}\ t\ \texttt{in}\ \lambda u.t$.

$$\dfrac{\dfrac{\dfrac{}{\lambda x.x \Downarrow_e \lambda x.x}(\mathrm{Ax}) \quad \dfrac{\dfrac{}{\lambda y.y \Downarrow_e \lambda y.y}(\mathrm{Ax}) \quad \dfrac{}{z\{z \leftarrow \lambda y.y\} \Downarrow_e \lambda y.y}(\mathrm{Ax})}{\texttt{let}\ \lambda y.y\ \texttt{be}\ z\ \texttt{in}\ z \Downarrow_e \lambda y.y}(\text{Let-var})}{M_1 \Downarrow_e \langle \lambda x.x, \lambda y.y \rangle \equiv M_2}(\text{Pair}) \quad \dfrac{}{(\lambda u.t)\{t \leftarrow M_2\} \Downarrow_e \lambda u.M_2}(\mathrm{Ax})}{M \Downarrow_e \lambda u.\langle \lambda x.x, \lambda y.y \rangle}$$

Propriété 3.17 Si M est un terme clos de TPC et si $M \Downarrow_{e,TPC} K$ alors K est une fns et si K est une fns alors $K \Downarrow_{e,TPC} K$.

Preuve:
On démontre sans difficulté cette propriété par récurrence sur la taille de l'arbre de preuve de $M \Downarrow_e K$.

$\square$

L'évaluateur $\Downarrow_{e,TPC}$ est déterministe comme le montre le théorème suivant.

Théorème 3.18 (Déterminisme de $\Downarrow_{e,TPC}$) $\Downarrow_{e,TPC}$ est déterministe.

Preuve:
On démontre ce théorème par la même technique que pour le théorème (3.2).

$\square$

La proposition suivante montre la correction de $\Downarrow_{e,TPC}$ par rapport à TPC.

Proposition 3.19 Si M est un terme clos de TPC et si $M \Downarrow_{e,TPC} K$ alors $\mathrm{M} \rightarrow_{tpc}^{*} \mathrm{K}$.

Preuve:
Similaire à celle de la proposition (3.3).

$\square$

La relation $\Downarrow_{e,TPC}$ sur les termes de TPC préserve les types.

Théorème 3.20 (Préservation de types pour $\Downarrow_{e,TPC}$) Si M est un terme clos de TPC et si $\triangleright M : A$ et $M \Downarrow_{e,TPC} K$ alors $\triangleright K : A$.

Preuve:
Ce théorème est une application immédiate de la proposition (3.19) et du théorème (5.1) de [4].

$\square$

3.4 Un évaluateur strict pour TPC_{ES}

Nous définissons ici l'évaluateur strict d'un grand pas pour TPC_{ES} que nous noterons $\Downarrow_{e,TPC_{ES}}$ (ou $\Downarrow_e$ quand toute confusion sera impossible). Cet évaluateur implante la stratégie stricte pour TPC_{ES}. Il possède des propriétés similaires à celle de $\Downarrow_{l,TPC_{ES}}$ à savoir le déterminisme, la correction par rapport à TPC_{ES}, l'équivalence avec $\Downarrow_{e,TPC}$ et la préservation de type.

Définition 3.7 (L'évaluateur $\Downarrow_{e,TPC_{ES}}$) La relation $\Downarrow_{e,TPC_{ES}}$ est une relation entre les termes clos et purs de TPC_{ES} et les fns. La relation $\rightsquigarrow_e$, qui est le "purificateur" de cette stratégie est une relation entre termes quelconques de TPC_{ES}.

Dans ces règles, $\mathrm{f} \in \{\texttt{let \_ be } P \texttt{ in \_}; \langle \_,\_ \rangle; \lambda P.\_; \texttt{inr}(\_); \texttt{inl}(\_)\}$.

$$\frac{\texttt{let }(\texttt{let } M \texttt{ be } P \texttt{ in } J) \texttt{ be } Q \texttt{ in } N \Downarrow_{e,TPC_{ES}} K}{\lambda P.J \texttt{ of } M \texttt{ is } Q \texttt{ in } N \Downarrow_{e,TPC_{ES}} K} \text{ (Of)}$$

$$\frac{M \Downarrow_{e,TPC_{ES}} \langle M_1, M_2 \rangle \qquad \texttt{let } M_1 \texttt{ be } P_1 \texttt{ in } (\texttt{let } M_2 \texttt{ be } P_2 \texttt{ in } N) \Downarrow_{e,TPC_{ES}} K}{\texttt{let } M \texttt{ be } \langle P_1, P_2 \rangle \texttt{ in } N \Downarrow_{e,TPC_{ES}} K} \text{ (Prod)}$$

$$\frac{\texttt{let } M \texttt{ be } P_1 \texttt{ in let } J \texttt{ be } P_2 \texttt{ in } N \Downarrow_{e,TPC_{ES}} K}{\texttt{let } M \texttt{ be } @(P_1, P_2) \texttt{ in } N \Downarrow_{e,TPC_{ES}} K} \text{ (@)}$$

$$\frac{M \Downarrow_{e,TPC_{ES}} \texttt{inl}(L) \qquad N[\xi/L] \rightsquigarrow_e N_1 \qquad \texttt{let } L \texttt{ be } P \texttt{ in } N_1 \Downarrow_{e,TPC_{ES}} K}{\texttt{let } M \texttt{ be } (P|_\xi Q) \texttt{ in } N \Downarrow_{e,TPC_{ES}} K} \text{ (Let-left)}$$

$$\frac{M \Downarrow_{e,TPC_{ES}} \texttt{inr}(L) \qquad N[\xi/R] \rightsquigarrow_e N_1 \qquad \texttt{let } L \texttt{ be } Q \texttt{ in } N_1 \Downarrow_{e,TPC_{ES}} K}{\texttt{let } M \texttt{ be } (P|_\xi Q) \texttt{ in } N \Downarrow_{e,TPC_{ES}} K} \text{ (Let-right)}$$

$$\frac{M \Downarrow_{e,TPC_{ES}} \lambda P.J \qquad N[z/\lambda P.J] \rightsquigarrow_e N_1 \qquad N_1 \Downarrow_{e,TPC_{ES}} K}{\texttt{let } M \texttt{ be } \#z \texttt{ in } N \Downarrow_{e,TPC_{ES}} K} \text{ (Let-lambda)}$$

$$\frac{M \Downarrow_{e,TPC_{ES}} J \qquad N[x/J] \rightsquigarrow_e N_1 \qquad N_1 \Downarrow_{e,TPC_{ES}} K}{\texttt{let } M \texttt{ be } x \texttt{ in } N \Downarrow_{e,TPC_{ES}} K} \text{ (Let-var)}$$

$$\frac{N \Downarrow_{e,TPC_{ES}} K}{\texttt{let } M \texttt{ be \_ in } N \Downarrow_{e,TPC_{ES}} K} \text{ (Let-weak)}$$

$$\frac{M \Downarrow_{e,TPC_{ES}} K}{[M|_{\mathrm{L}} N] \Downarrow_{e,TPC_{ES}} K} \text{ (Left)} \qquad \frac{N \Downarrow_{e,TPC_{ES}} K}{[M|_{\mathrm{R}} N] \Downarrow_{e,TPC_{ES}} K} \text{ (Right)}$$

$$\frac{M \Downarrow_{e,TPC_{ES}} L}{\texttt{inl}(M) \Downarrow_{e,TPC_{ES}} \texttt{inl}(L)} \text{ (Inl)} \qquad \frac{M \Downarrow_{e,TPC_{ES}} L}{\texttt{inr}(M) \Downarrow_{e,TPC_{ES}} \texttt{inr}(L)} \text{ (Inr)}$$

$$\frac{M \Downarrow_{e,TPC_{ES}} K \qquad N \Downarrow_{e,TPC_{ES}} L}{\langle M, N \rangle \Downarrow_{e,TPC_{ES}} \langle K, L \rangle} \text{ (Pair)}$$

$$\frac{}{\lambda P.M \Downarrow_{e,TPC_{ES}} \lambda P.M} \text{ (Ax)}$$

$$\frac{}{x[x/M] \rightsquigarrow_e M} \qquad \frac{}{y[\theta] \rightsquigarrow_e y} \prec \text{ si } y \neq Dom(\theta)$$

$$\frac{M_1[\theta] \rightsquigarrow_e K_1 \qquad \ldots \qquad M_n[\theta] \rightsquigarrow_e K_n}{f(M_1, \ldots, M_n)[\theta] \rightsquigarrow_e f(K_1, \ldots, K_n)}$$

$$\frac{}{(z \texttt{ of } M \texttt{ is } P \texttt{ in } N)[z/L] \rightsquigarrow_e L \texttt{ of } M \texttt{ is } P \texttt{ in } N}$$

$$\frac{M[\theta] \rightsquigarrow_e M_1 \qquad N[\theta] \rightsquigarrow_e N_1}{(z \texttt{ of } M \texttt{ is } P \texttt{ in } N)[\theta] \rightsquigarrow_e z \texttt{ of } M_1 \texttt{ is } P \texttt{ in } N_1} \prec \text{ si } z \neq Dom(\theta)$$

$$\frac{J[\theta] \rightsquigarrow_e J_1 \qquad M[\theta] \rightsquigarrow_e M_1 \qquad N[\theta] \rightsquigarrow_e N_1}{(\lambda Q.J \texttt{ of } M \texttt{ is } P \texttt{ in } N)[\theta] \rightsquigarrow_e \lambda Q.J_1 \texttt{ of } M_1 \texttt{ is } P \texttt{ in } N_1}$$

$$\frac{M[\theta] \rightsquigarrow_e M_1 \qquad N[\theta] \rightsquigarrow_e N_1}{[M|_{\mathrm{T}} N][\theta] \rightsquigarrow_e [M_1|_{\mathrm{T}} N_1]} \prec \text{ si } T \neq Dom(\theta)$$

$$\frac{M[\theta] \rightsquigarrow_e M_1 \qquad N[\theta] \rightsquigarrow_e N_1}{[M|_\xi N][\xi/\mathtt{K}] \rightsquigarrow_e [M_1|_\mathtt{K} N_1]}$$

Remarque 3.21 On remarque que la relation $\rightsquigarrow_e$ est nécessaire car elle est une relation entre des termes non clos alors que les évaluateurs sont toujours définis sur des termes clos.

Nous donnons maintenant un exemple d'évaluation.. soit M_1 le terme de TPC_{ES} suivant: $M_1 = \langle \lambda x.x, \mathtt{let}\ \lambda y.y\ \mathtt{be}\ z\ \mathtt{in}\ z\rangle$, et $M = \mathtt{let}\ M_1\ \mathtt{be}\ t\ \mathtt{in}\ \lambda u.t$.
Nous avons découpé l'évaluation en trois arbres pour des raison de place.

$$\dfrac{\dfrac{}{\lambda y.y \Downarrow_e \lambda y.y}(\mathrm{Ax}) \qquad \dfrac{}{z[z/\lambda y.Y] \rightsquigarrow_e \lambda y.y}(\mathrm{Es\text{-}Ax1}) \qquad \dfrac{}{\lambda y.y \Downarrow_e \lambda y.Y}(\mathrm{Ax})}{\mathtt{let}\ \lambda y.y\ \mathtt{be}\ z\ \mathtt{in}\ z \Downarrow_e \lambda y.y}(\mathrm{Let\text{-}Var})$$

$$\dfrac{\dfrac{}{\lambda x.x \Downarrow_e \lambda x.x}(\mathrm{Ax}) \qquad \mathtt{let}\ \lambda y.y\ \mathtt{be}\ z\ \mathtt{in}\ z \Downarrow_e \lambda y.y}{M_1 \Downarrow_e \langle \lambda x.x, \lambda y.y\rangle \equiv M_2}(\mathrm{Pair})$$

$$\dfrac{M_1 \Downarrow_e \langle \lambda x.x, \lambda y.y\rangle \equiv M_2 \qquad \dfrac{\dfrac{}{t[t/M_2] \rightsquigarrow_e M_2}(\mathrm{Es\text{-}ax1})}{(\lambda u.t)[t/M_2] \rightsquigarrow_e M_2} \qquad \dfrac{}{\lambda u.M_2 \Downarrow_e \lambda u.M_2}(\mathrm{Ax})}{M \Downarrow_e \lambda u.M_2}(\mathrm{Let\text{-}var})$$

Propriété 3.22

1. Si M et L sont purs alors si K est tel que $M[x/L] \rightsquigarrow_e K$ alors K $\equiv$ ES($M[x/L]$) .
2. Si M est pur et si K est tel que $M[\xi/T] \rightsquigarrow_e K$ alors K $\equiv M\{\xi \leftarrow T\}$
3. Si M est pur et clos et si $M \Downarrow_{e,TPC_{ES}} K$ alors K est une fns. Si K est une fns alors $K \Downarrow_{e,TPC_{ES}} K$.

<u>Preuve:</u>
Par une simple récurrence sur la taille de l'arbre de preuve de $M[x/L] \rightsquigarrow_e K$, $M[\xi/T] \rightsquigarrow_e K$ et de $M \Downarrow_e K$ respectivement.

□

Lemme 3.23 Si M est un terme clos et pur de TPC_{ES} et si $M \Downarrow_{e,TPC_{ES}} K$ alors K est pur.

<u>Preuve:</u>
Par récurrence sur la taille de la preuve de $M \Downarrow_e K$.
<u>Cas de base:</u> $M \Downarrow_e M$ le résultat est trivial.
<u>Cas inductif:</u>

- Si $M \equiv \lambda P.J\ \mathtt{of}\ M_1\ \mathtt{is}\ Q\ \mathtt{in}\ N \Downarrow_e K$ avec $R = \mathtt{let}\ (\mathtt{let}\ M_1\ \mathtt{be}\ P\ \mathtt{in}\ J)\ \mathtt{be}\ Q\ \mathtt{in}\ N \Downarrow_e K$ alors R est clairement pur donc par l'hypothèse de récurrence K est pur. Tous les cas qui ne font pas apparaître de nouvelle substitution se traitent exactement de la même manière.

– Si $M \equiv \texttt{let}\ M_1\ \texttt{be}\ (P \mid_\xi Q)\ \texttt{in}\ N \Downarrow_e K$ avec $M_1 \Downarrow_e \texttt{inl}(L)$, $N[\xi/\texttt{L}] \rightsquigarrow_e N_1$ et $\texttt{let}\ L\ \texttt{be}\ P\ \texttt{in}\ N_1 \Downarrow_e K$ alors par hypothèse de récurrence L est pur, par la propriété (3.22-1) N_1 est pur et enfin par hypothèse de récurrence K est pur. Tous les cas faisant apparaître une substitution explicite se traitent par une méthode similaire.

□

Le théorème suivant montre l'équivalence annoncée entre $\Downarrow_{e,TPC}$ et $\Downarrow_{e,TPC_{ES}}$.

Théorème 3.24 (Equivalence entre $\Downarrow_{e,TPC}$et $\Downarrow_{e,TPC_{ES}}$) Si M est un terme clos et pur de TPC_{ES} alors $M \Downarrow_{e,TPC_{ES}} K \Leftrightarrow M \Downarrow_{e,TPC} K$.

<u>Preuve:</u>
Par récurrence sur la taille de l'arbre de preuve de $M \Downarrow_{e,TPC} K$ ou de $M \Downarrow_{e,TPC_{ES}} K$ suivant le sens considéré.
<u>Cas de base:</u> Si $M \Downarrow_{e,TPC} M$ alors par la propriété (3.22-2) M est une fnp et on a bien $M \Downarrow_{e,TPC_{ES}} M$. De même si $M \Downarrow_{e,TPC_{ES}} K$
<u>Cas inductif:</u>

– Tous les cas ne faisant pas apparaître de substitution se traitent trivialement par récurrence.

– Pour les cas faisant apparaître une substitution on applique l'hypothèse de récurrence et la propriété (3.22-1).

□

Théorème 3.25 (Déterminisme de $\Downarrow_{e,TPC_{ES}}$) $\Downarrow_{e,TPC_{ES}}$ est déterministe.

<u>Preuve:</u>
Par application du lemme (3.24) et du théorème (3.18).

□

L' évaluateur $\Downarrow_{e,TPC_{ES}}$ est correct par rapport à TPC_{ES}.

Proposition 3.26 Si M est un terme clos et pur de TPC_{ES} et si $M \Downarrow_{e,TPC_{ES}} K$ alors M $\rightarrow^*_{tpces}$ K.

<u>Preuve:</u>
Par le lemme (3.24) et la proposition (3.19) et le lemme (6.2) dans [4].

□

La relation $\Downarrow_{e,TPC_{ES}}$ sur les termes de TPC préserve les types comme le montre le théorème suivant.

Théorème 3.27 (Préservation de types pour $\Downarrow_{e,TPC_{ES}}$) Si M est un terme clos et pur de TPC_{ES} et si $\triangleright M : A$ et $M \Downarrow_{e,TPC_{ES}} K$ alors $\triangleright K : A$.

<u>Preuve:</u>
Ce théorème est une application immédiate de la proposition (3.26) et du théorème (3.8) de [4].

□

4 Les évaluateurs pas à pas

Contrairement aux évaluateurs d'un grand pas, les évaluateurs pas à pas ne donnent pas comme résultat d'un calcul la "valeur" d'un programme mais, comme leur nom l'indique, le résultat d'un pas de calcul. Pour chacun des quatre évaluateurs pas à pas que nous allons définir nous montrerons l'équivalence avec l'évaluateur d'un grand pas correspondant. Nous donnerons aussi des propriétés propres aux évaluateurs pas à pas afin de prouver certaines propriétés des deux stratégies paresseuse et stricte.

4.1 Un évaluateur paresseux pour TPC

L'évaluateur $\Rightarrow_{l,TPC}$ implante la stratégie paresseuse pas à pas pour TPC. Nous allons montrer son équivalence avec $\Downarrow_{l,TPC}$ dans le sens où nous allons montrer que si M est un terme de TPC et si K est une fnp alors $M \Downarrow_{l,TPC} K \Leftrightarrow M \Rightarrow^*_{l,TPC} K$. La relation $\Rightarrow_{l,TPC}$ préserve aussi les types. Nous montrerons aussi que tout terme clos et bien typé de TPC est soit une fnp soit réductible par $\Rightarrow_{l,TPC}$. Ceci nous permettra alors de conclure que tout terme clos et bien typé de TPC est évaluable par $\Downarrow_{l,TPC}$.

Définition 4.1 La relation $\Rightarrow_{l,TPC}$ est une relation définie sur les termes clos de TPC.

$$\frac{}{\lambda P.J\ \texttt{of}\ M\ \texttt{is}\ Q\ \texttt{in}\ N \Rightarrow_{l,TPC} \texttt{let}\ (\texttt{let}\ M\ \texttt{be}\ P\ \texttt{in}\ J)\ \texttt{be}\ Q\ \texttt{in}\ N}$$

$$\frac{}{\texttt{let}\ \langle M_1,M_2\rangle\ \texttt{be}\ \langle P_1,P_2\rangle\ \texttt{in}\ N \Rightarrow_{l,TPC} \texttt{let}\ M_1\ \texttt{be}\ P_1\ \texttt{in}\ (\texttt{let}\ M_2\ \texttt{be}\ P_2\ \texttt{in}\ N)}$$

$$\frac{}{\texttt{let}\ M\ \texttt{be}\ @(P_1,P_2)\ \texttt{in}\ N \Rightarrow_{l,TPC} \texttt{let}\ M\ \texttt{be}\ P_1\ \texttt{in}\ (\texttt{let}\ M\ \texttt{be}\ P_2\ \texttt{in}\ N)}$$

$$\frac{}{\texttt{let}\ \texttt{inl}(L)\ \texttt{be}\ (P|_\xi Q)\ \texttt{in}\ N \Rightarrow_{l,TPC} \texttt{let}\ L\ \texttt{be}\ P\ \texttt{in}\ N\{\xi \leftarrow \texttt{L}\}}$$

$$\frac{}{\texttt{let}\ \texttt{inr}(L)\ \texttt{be}\ (P|_\xi Q)\ \texttt{in}\ N \Rightarrow_{l,TPC} \texttt{let}\ L\ \texttt{be}\ Q\ \texttt{in}\ N\{\xi \leftarrow \texttt{R}\}}$$

$$\frac{}{\texttt{let}\ \lambda P.J\ \texttt{be}\ \#z\ \texttt{in}\ N \Rightarrow_{l,TPC} N\{z \leftarrow \lambda P.J\}}$$

$$\frac{}{\texttt{let}\ M\ \texttt{be}\ x\ \texttt{in}\ N \Rightarrow_{l,TPC} N\{x \leftarrow M\}}$$

$$\frac{}{\texttt{let}\ M\ \texttt{be}\ \_\ \texttt{in}\ N \Rightarrow_{l,TPC} N}$$

$$\frac{}{[M|_{\texttt{L}}N] \Rightarrow_{l,TPC} M}$$

$$\frac{}{[M|_{\texttt{R}}N] \Rightarrow_{l,TPC} N}$$

$$\frac{M \Rightarrow_{l,TPC} M_1}{\texttt{let}\ M\ \texttt{be}\ P\ \texttt{in}\ N \Rightarrow_{l,TPC} \texttt{let}\ M_1\ \texttt{be}\ P\ \texttt{in}\ N} \prec \text{si on ne peut appliquer une des autres règles (Gen)}$$

La règle (Gen) exprime le fait que pour évaluer un terme de la forme `let` M `be` P `in` N, il faut commencer par "mettre en adéquation" le sous-terme M avec le motif P. C'est en fait la règle du filtrage explicite de cet évaluateur.
La relation $\Rightarrow_{l,TPC}$ étant un évaluateur pas à pas, les règles (Ax_i) de la relation $\Downarrow_{l,TPC}$ ont été purement et simplement supprimées de la relation $\Rightarrow_{l,TPC}$ par rapport à la relation $\Downarrow_{l,TPC}$.

Remarque 4.1 On remarque que $\Rightarrow_{l,TPC} \subset \rightarrow_{tpc}$. En fait les relations $\Rightarrow_{l,TPC}$ et $\rightarrow_{tpc}$ se ressemblent beaucoup. Leur principale différence est que $\Rightarrow_{l,TPC}$ "déterminise" la relation $\rightarrow_{tpc}$.

Nous allons donner maintenant la réduction du terme M défini comme suit: $M_1 = \langle\ \lambda x.x,\ \texttt{let}\ \lambda y.y\ \texttt{be}\ z\ \texttt{in}\ z\ \rangle$, et $M = \texttt{let}\ M_1\ \texttt{be}\ t\ \texttt{in}\ \lambda u.t$.

$M \Rightarrow_{l,TPC} (\lambda u.t)\{t \leftarrow M_1\} \equiv \lambda u.M_1$

Les deux propositions suivantes montrent l'équivalence entre $\Downarrow_{l,TPC}$ et $\Rightarrow_{l,TPC}$.

Proposition 4.2 (Equivalence entre $\Downarrow_{l,TPC}$et $\Rightarrow_{l,TPC}$ (I)) Si M est un terme clos de TPC et si $M \Downarrow_{l,TPC} K$ alors $M \Rightarrow^*_{l,TPC} K$ et même $M \Rightarrow^+_{l,TPC} K$ si M n'est pas une fnp.

<u>Preuve:</u>
Par récurrence sur la taille de l'arbre de preuve de $M \Downarrow_{l,TPC} K$.
<u>Cas de base:</u> Si $M \Downarrow_l M$ alors on a trivialement $M \Rightarrow^*_{l,TPC} M$.
<u>Cas inductif:</u>

- $M \equiv \lambda\ P.\ J\ \texttt{of}\ L\ \texttt{is}\ Q\ \texttt{in}\ N \Downarrow_l K$ provient de $R = \texttt{let (let}\ L\ \texttt{be}\ P\ \texttt{in}\ J)\ \texttt{be}\ Q\ \texttt{in}\ N \Downarrow_l K$. Alors par hypothèse de récurrence on a $R \Rightarrow^*_{l,TPC} K$. De plus on a $\lambda P.J\ \texttt{of}\ L\ \texttt{is}\ Q\ \texttt{in}\ N \Rightarrow_{l,TPC} R$ donc on a bien $M \Rightarrow^+_{l,TPC} K$.
- $M \equiv \texttt{let}\ J\ \texttt{be}\ \langle P_1, P_2\rangle\ \texttt{in}\ N \Downarrow_l K$ provient de $J \Downarrow_l \langle M_1, M_2\rangle$ et $R = \texttt{let}\ M_1\ \texttt{be}\ P_1\ \texttt{in (let}\ M_2\ \texttt{be}\ P_2\ \texttt{in}\ N) \Downarrow_l K$. Alors par hypothèse de récurrence on a: $J \Rightarrow^*_{l,TPC} \langle M_1, M_2\rangle$ et $R \Rightarrow^*_{l,TPC} K$. De plus on a $M \Rightarrow^*_{l,TPC} \texttt{let}\ \langle M_1, M_2\rangle\ \texttt{be}\ \langle P_1, P_2\rangle\ \texttt{in}\ N$. Enfin on a: $\texttt{let}\ \langle M_1, M_2\rangle\ \texttt{be}\ \langle P_1, P_2\rangle\ \texttt{in}\ N \Rightarrow_{l,TPC} R$. Donc on a bien: $M \Rightarrow^+_{l,TPC} K$.
- $M \equiv \texttt{let}\ J\ \texttt{be}\ @(P_1, P_2)\ \texttt{in}\ N \Downarrow_l K$ provient de $R = \texttt{let}\ J\ \texttt{be}\ P_1\ \texttt{in (let}\ J\ \texttt{be}\ P_2\ \texttt{in}\ N) \Downarrow_l K$. Par hypothèse de récurrence on a $R \Rightarrow^*_{l,TPC} K$. De plus on a: $M \Rightarrow_{l,TPC} R$. Donc on a bien $M \Rightarrow^+_{l,TPC} K$.
- $M \equiv \texttt{let}\ J\ \texttt{be}\ (P\ |_\xi\ Q)\ \texttt{in}\ N \Downarrow_l K$ provient de $J \Downarrow_l \texttt{inl}(L)$ et $R = \texttt{let}\ L\ \texttt{be}\ P\ \texttt{in}\ N\{\xi \leftarrow \mathrm{L}\} \Downarrow_l K$. Par hypothèse de récurrence on a: $J \Rightarrow^*_{l,TPC} \texttt{inl}(L)$ et $R \Rightarrow^*_{l,TPC} K$. De plus on a clairement $M \Rightarrow^*_{l,TPC} \texttt{let inl}(L)\ \texttt{be}\ (\mathrm{P}|_\xi \mathrm{Q})\ \texttt{in}\ N$. On a: $\texttt{let inl}(L)\ \texttt{be}\ (P\ |_\xi\ Q)\ \texttt{in}\ N \Rightarrow_{l,TPC} R$. On a donc bien: $M \Rightarrow^+_{l,TPC} K$. Le cas où $J \Downarrow_l \texttt{inr}(L)$ est similaire à celui-ci.
- $M \equiv \texttt{let}\ L\ \texttt{be}\ \#z\ \texttt{in}\ N \Downarrow_l K$ provient de $L \Downarrow_l \lambda P.J$ et $N\{z \leftarrow \lambda P.J\} \Downarrow_l K$. Alors par hypothèse de récurrence on a $L \Rightarrow^*_{l,TPC} \lambda P.J$ et $N\{z \leftarrow \lambda P.J\} \Rightarrow^*_{l,TPC} K$. De plus on a: $M \Rightarrow^*_{l,TPC} \texttt{let}\ \lambda\ P.\ J\ \texttt{be}\ \#z\ \texttt{in}\ N$ et $\texttt{let}\ \lambda\ P.\ J\ \texttt{be}\ \#z\ \texttt{in}\ N \Rightarrow_{l,TPC} N\{z \leftarrow \lambda P.J\}$ donc on a bien $M \Rightarrow^+_{l,TPC} K$.
- $M \equiv \texttt{let}\ J\ \texttt{be}\ x\ \texttt{in}\ N \Downarrow_l K$ provient de $N\{x \leftarrow J\} \Downarrow_l K$ Alors par hypothèse de récurrence on a $N\{x \leftarrow J\} \Rightarrow^*_{l,TPC} K$. De plus on a: $M \Rightarrow_{l,TPC} N\{x \leftarrow J\}$, donc on a bien: $M \Rightarrow^+_{l,TPC} K$.

- $M \equiv \texttt{let}\ J\ \texttt{be}\ \_\ \texttt{in}\ N \Downarrow_l K$ provient de $N \Downarrow_l K$. Alors par hypothèse de récurrence on a $N \Rightarrow^*_{l,TPC} K$. De plus on a: $M \Rightarrow_{l,TPC} N$, donc on a bien: $M \Rightarrow^+_{l,TPC} K$.

- $[M_1 |_{\mathtt{L}} M_2] \Downarrow_l K$ provient de $M_1 \Downarrow_l K$. Alors par hypothèse de récurrence $M_1 \Rightarrow^*_{l,TPC} K$. De plus on a $[M_1 |_{\mathtt{L}} M_2] \Rightarrow_{l,TPC} M_1$, donc on a bien $M \Rightarrow^+_{l,TPC} K$.

- $[M_1 |_{\mathtt{R}} M_2] \Downarrow_l K$ provient de $M_2 \Downarrow_l K$. Alors par hypothèse de récurrence $M_2 \Rightarrow^*_{l,TPC} K$. De plus on a $[M_1 |_{\mathtt{R}} M_2] \Rightarrow_{l,TPC} M_2$, donc on a bien $M \Rightarrow^+_{l,TPC} K$.

□

Proposition 4.3 (Equivalence entre $\Downarrow_{l,TPC}$ et $\Rightarrow_{l,TPC}$ (II)) Si M est un terme clos de TPC et si $M \Rightarrow^*_{l,TPC} K$ et K est une fnp alors $M \Downarrow_{l,TPC} K$.

Preuve:
Par récurrence sur la longueur de la dérivation de $M \Rightarrow^*_{l,TPC} K$
Cas de base: Si M se réduit par $\Rightarrow_{l,TPC}$ en 0 pas en K alors M est une fnp et par la propriété (3.1) $M \Downarrow_l M$.
Cas inductif: M se réduit en K par $\Rightarrow_{l,TPC}$ en $n > 0$ pas

- $M \equiv [M_1 |_{\mathtt{L}} M_2] \Rightarrow^+_{l,TPC} K$ alors on a $M \Rightarrow_{l,TPC} M_1 \Rightarrow^*_{l,TPC} K$ et donc par hypothèse de récurrence $M_1 \Downarrow_l K$ donc on a bien $M \Downarrow_l K$. Idem si M $\equiv [M_1 |_{\mathtt{R}} M_2]$.

- $M \equiv \texttt{let}\ \langle M_1, M_2\rangle\ \texttt{be}\ \langle P_1, P_2\rangle\ \texttt{in}\ N \Rightarrow^+_{l,TPC} K$ alors on a: $M \Rightarrow_{l,TPC} R = \texttt{let}\ M_1\ \texttt{be}\ P_1\ \texttt{in}\ (\texttt{let}\ M_2\ \texttt{be}\ P_2\ \texttt{in}\ N) \Rightarrow^*_{l,TPC} K$ et donc par hypothèse de récurrence on a $R \Downarrow_l K$ et donc on a bien $M \Downarrow_l K$.

- $M \equiv \texttt{let}\ J\ \texttt{be}\ @(P,Q)\ \texttt{in}\ N \Rightarrow^+_{l,TPC} K$ alors on a: $M \Rightarrow_{l,TPC} R = \texttt{let}\ J\ \texttt{be}\ P_1\ \texttt{in}\ (\texttt{let}\ J\ \texttt{be}\ P_2\ \texttt{in}\ N) \Rightarrow^*_{l,TPC} K$ et donc par hypothèse de récurrence on a $R \Downarrow_l K$. On a donc bien $M \Downarrow_l K$.

- $M \equiv \texttt{let}\ \texttt{inl}(L)\ \texttt{be}\ (P |_\xi Q)\ \texttt{in}\ N \Rightarrow^+_{l,TPC} K$ on a alors: $M \Rightarrow_{l,TPC} R = \texttt{let}\ L\ \texttt{be}\ P\ \texttt{in}\ N\{\xi \leftarrow \mathtt{L}\} \Rightarrow^*_{l,TPC} K$ donc par hypothèse de récurrence on a $R \Downarrow_l K$ et donc $M \Downarrow_l K$. Idem si M $\equiv \texttt{let}\ \texttt{inr}(L)\ \texttt{be}$ $(\mathrm{P}|_\xi\mathrm{Q})\ \texttt{in}$ N.

- $M \equiv \texttt{let}\ J\ \texttt{be}\ P\ \texttt{in}\ N \Rightarrow^+_{l,TPC} K$ provient de $M \Rightarrow^k_{l,TPC} R = \texttt{let}\ K_1\ \texttt{be}\ P\ \texttt{in}\ N \Rightarrow^+_{l,TPC} K$ avec $J \Rightarrow^k_{l,TPC} K_1$ et K_1 est une fnpet $n > k > 0$. On remarque que l'on peut toujours atteindre un K_1 qui est une fnp puisque la règle (Gen) ne s'applique que si et seulement si J n'est pas une fnp. Par hypothèse de récurrence, on a $R \Downarrow_l K$ et $J \Downarrow_l K_1$ et donc on a bien $M \Downarrow_l K$.

- Les autres cas se traitent de manière similaire au précédent.

□

Le lemme suivant montre que tout terme bien typé de TPC est soit une fnp (donc un résultat pour la relation $\Downarrow_{l,TPC}$) soit réductible par $\Rightarrow_{l,TPC}$.

Lemme 4.4 Si M est un programme de TPC et si M n'est pas une fnp alors $M \Rightarrow_{l,TPC} N$ pour un certain N.

Preuve:
Par induction sur la structure de M.
On remarque que M étant clos les cas où $M \equiv x$ et $M \equiv z$ **of** M_1 **is** P **in** N sont impossibles.

- Si M $\equiv \lambda P.J$ `of` M_1 `is` Q `in` N on a: $M \Rightarrow_{l,TPC}$ `let` (`let` M_1 `be` P `in` J) `be` Q `in` N.
- Tout les autres cas se traitent de la même manière sauf le cas où M $\equiv$ `let` M_1 `be` P `in` N et où aucune autre règle que la dernière ne s'applique. Dans ce cas, par hypothèse d'induction $M_1 \Rightarrow_{l,TPC} M_2$ et donc $M \Rightarrow_{l,TPC}$ `let` M_2 `be` P `in` N.

$\square$

On tire en particulier de la remarque (4.1) que, comme $\rightarrow_{tpc}$, $\Rightarrow_{l,TPC}$ a la propriété de préserver les types.

Le théorème suivant montre la "complétude" de $\Downarrow_{l,TPC}$ pour TPC.

Théorème 4.5 Si M est un programme de TPC alors $M \Downarrow_{l,TPC} K$ pour un certain K.

Preuve:
Par le lemme (4.4) $M \Rightarrow^*_{l,TPC} K$ pour une certaine fnp K et par la proposition (4.3) $M \Downarrow_l K$.

$\square$

4.2 Un évaluateur paresseux pour TPC_{ES}

L'évaluateur $\Rightarrow_{l,TPC_{ES}}$ implante la stratégie paresseuse pas à pas pour TPC_{ES}. Il est équivalent à $\Downarrow_{l,TPC_{ES}}$ et hérite donc de ses propriétés.

Définition 4.2 La relation $\Rightarrow_{l,TPC_{ES}}$ est une relation entre les termes clos de TPC_{ES} tout comme la relation $\nearrow$. Les deux relations $\tilde{\rightsquigarrow}$ et $\tilde{\hookrightarrow}$ sont des relations entre les termes quelconques de TPC_{ES}. Les quatre relations ont les même fonctions que les quatre relations correspondantes pour l'évaluateur d'un grand pas $\Downarrow_{l,TPC_{ES}}$ à savoir: $\nearrow$ est la relation qui effectue en réalité le calcul, $\tilde{\rightsquigarrow}$ se contente de faire descendre les substitution de la tête d'un terme, $\tilde{\hookrightarrow}$ est un purificateur et $\Rightarrow_{l,TPC_{ES}}$ est l'évaluateur en lui-même.

Dans ces règles:
f $\in$ { `let` _ `be` P `in` _;$\langle$_, _$\rangle$;λP._;`inr`(_);`inl`(_) }
g est n'importe quel constructeur autre que la substitution.

$$\frac{}{\lambda P.J \ \texttt{of}\ M \ \texttt{is}\ Q \ \texttt{in}\ N \nearrow \texttt{let}\ (\texttt{let}\ M \ \texttt{be}\ P \ \texttt{in}\ J) \ \texttt{be}\ Q \ \texttt{in}\ N}$$

$$\frac{}{\texttt{let}\ \langle M_1, M_2\rangle \ \texttt{be}\ \langle P_1, P_2\rangle \ \texttt{in}\ N \nearrow \texttt{let}\ M_1 \ \texttt{be}\ P_1 \ \texttt{in}\ (\texttt{let}\ M_2 \ \texttt{be}\ P_2 \ \texttt{in}\ N)}$$

$$\frac{}{\texttt{let}\ M \ \texttt{be}\ @(P_1, P_2) \ \texttt{in}\ N \nearrow \texttt{let}\ M \ \texttt{be}\ P_1 \ \texttt{in}\ (\texttt{let}\ M \ \texttt{be}\ P_2 \ \texttt{in}\ N)}$$

$$\frac{}{\texttt{let}\ inl(L) \ \texttt{be}\ (P|_\xi Q) \ \texttt{in}\ N \nearrow \texttt{let}\ L \ \texttt{be}\ P \ \texttt{in}\ N[\xi/L]}$$

$$\frac{}{\texttt{let}\ inr(L) \ \texttt{be}\ (P|_\xi Q) \ \texttt{in}\ N \nearrow \texttt{let}\ L \ \texttt{be}\ Q \ \texttt{in}\ N[\xi/R]}$$

$$\frac{}{\texttt{let}\ \lambda P.J \ \texttt{be}\ \#z \ \texttt{in}\ N \nearrow N[z/\lambda P.J]}$$

$$\frac{}{\texttt{let}\ M \ \texttt{be}\ x \ \texttt{in}\ N \nearrow N[x/M]}$$

$$\frac{}{\texttt{let}\ M \ \texttt{be}\ \_ \ \texttt{in}\ N \nearrow N}$$

$$\frac{}{[M|_{\mathtt{L}} N] \nearrow M}$$

$$\frac{}{[M|_{\mathtt{R}} N] \nearrow N}$$

$$\frac{M \nearrow M_1}{\texttt{let}\ M \ \texttt{be}\ P \ \texttt{in}\ N \nearrow \texttt{let}\ M_1 \ \texttt{be}\ P \ \texttt{in}\ N} \prec \text{Si aucune autre règle ne s'applique}$$

$$\frac{M[\theta] \tilde{\rightsquigarrow} M_1}{M[\theta]\Theta \nearrow M_1\Theta}$$

$$\frac{}{x[x/M] \tilde{\rightsquigarrow} M} \qquad \frac{}{y[\theta] \tilde{\rightsquigarrow} y} \prec \text{Si } y \neq Dom(\theta)$$

$$\frac{}{f(M_1, \cdots, M_n)[\theta] \tilde{\rightsquigarrow} f(M_1[\theta], \cdots, M_n[\theta])}$$

$$\frac{}{(z \ \texttt{of}\ M \ \texttt{is}\ P \ \texttt{in}\ N)[z/L] \tilde{\rightsquigarrow} L \ \texttt{of}\ M \ \texttt{is}\ P \ \texttt{in}\ N}$$

$$\frac{}{(z \ \texttt{of}\ M \ \texttt{is}\ P \ \texttt{in}\ N)[\theta] \tilde{\rightsquigarrow} z \ \texttt{of}\ M[\theta] \ \texttt{is}\ P \ \texttt{in}\ N[\theta]} \prec \text{Si } z \neq Dom(\theta)$$

$$\frac{}{(\lambda Q.J \ \texttt{of}\ M \ \texttt{is}\ P \ \texttt{in}\ N)[\theta] \tilde{\rightsquigarrow} \lambda Q.J[\theta] \ \texttt{of}\ M[\theta] \ \texttt{is}\ P \ \texttt{in}\ N[\theta]}$$

$$\frac{}{[M|_{\mathtt{T}} N][\theta] \tilde{\rightsquigarrow} [M[\theta]|_{\mathtt{T}} N[\theta]]} \prec \text{Si } \mathtt{T} \neq Dom(\theta) \qquad \frac{}{[M|_\xi N][\xi/\mathtt{K}] \tilde{\rightsquigarrow} [M[\xi/\mathtt{K}]|_{\mathtt{K}} N[\xi/\mathtt{K}]]}$$

$$\frac{}{x[x/M]\Theta \tilde{\hookrightarrow} M\Theta}$$

$$\frac{}{y[\theta]\Theta \tilde{\hookrightarrow} y\Theta} \prec \text{Si } y \neq Dom(\theta)$$

$$\frac{}{f(M_1, \ldots, M_n)\Theta \tilde{\hookrightarrow} f(M_1\Theta, \ldots, M_n\Theta)} \prec \text{Si } \Theta \text{ n'est pas vide}$$

$$\frac{M_{p+1} \tilde{\hookrightarrow} M^1_{p+1}}{g(K_1, \ldots, K_p, M_{p+1}, \ldots, M_n) \tilde{\hookrightarrow} g(K_1, \ldots, K_p, M^1_{p+1}, M_{p+2}, \ldots, M_n)} \prec \text{Si les } K_i \text{ sont purs et } M_{p+1} \text{ ne l'est pas}$$

$$\frac{}{(z \ \texttt{of}\ M \ \texttt{is}\ P \ \texttt{in}\ N)[z/L]\Theta \tilde{\hookrightarrow} (L \ \texttt{of}\ M \ \texttt{is}\ P \ \texttt{in}\ N)\Theta}$$

$$\frac{}{(z \ \texttt{of}\ M \ \texttt{is}\ P \ \texttt{in}\ N)[\theta]\Theta \tilde{\hookrightarrow} (z \ \texttt{of}\ M[\theta] \ \texttt{is}\ P \ \texttt{in}\ N[\theta])\Theta} \prec \text{Si z} \neq Dom(\theta)$$

$$\frac{}{(\lambda P.J \ \texttt{of}\ M \ \texttt{is}\ Q \ \texttt{in}\ N)\Theta \tilde{\hookrightarrow} (\lambda P.J\Theta \ \texttt{of}\ M\Theta \ \texttt{is}\ Q \ \texttt{in}\ N\Theta)} \prec \text{ Si } \Theta \text{ n'est pas vide}$$

$$\frac{}{[M|_{\mathtt{K}} N][\theta]\Theta \tilde{\hookrightarrow} [M[\theta]|_{\mathtt{K}} N[\theta]]\Theta} \prec \text{Si } \mathtt{K} \neq Dom(\theta)$$

$$\frac{}{[M|_\xi N][\xi/\mathtt{K}]\Theta \tilde{\hookrightarrow} [M[\xi/\mathtt{K}]\Theta|_{\mathtt{K}} N[\xi/\mathtt{K}]\Theta]}$$

$$\frac{M \nearrow M_1}{M \Rightarrow_{l,TPC_{ES}} M_1} \prec \text{si M n'est pas une fnp}^+$$

$$\frac{L \tilde{\hookrightarrow} L_1}{L \Rightarrow_{l,TPC_{ES}} L_1} \prec \text{si L est une fnp}^+$$

Nous allons maintenant donner un exemple de réduction. Pour cela on considère le terme M $M_1 = \langle \lambda x.x, \texttt{let}\ \lambda y.y\ \texttt{be}\ z\ \texttt{in}\ z \rangle$ et le terme $M = \texttt{let}\ M_1\ \texttt{be}\ t\ \texttt{in}\ \lambda u.t$.

$$\frac{M \nearrow (\lambda u.t)[t/M_1]}{M \Rightarrow_{l,TPC_{ES}} (\lambda u.t)[t/M_1]} \qquad \frac{\dfrac{(\lambda u.t)[t/M_1] \tilde{\rightsquigarrow} \lambda u.t[t/M_1]}{(\lambda u.t)[t/M_1] \nearrow \lambda u.t[t/M_1]}}{(\lambda u.t)[t/M_1] \Rightarrow_{l,TPC_{ES}} \lambda u.t[t/M_1]}$$

$$\frac{\dfrac{t[t/M_1] \tilde{\hookrightarrow} M_1}{\lambda u.t[t/M_1] \tilde{\hookrightarrow} \lambda u.M_1}}{\lambda u.t[t/M_1] \tilde{\hookrightarrow} \lambda u.M_1}$$

Le théorème suivant (4.9) montre l'équivalence entre $\Downarrow_{l,TPC_{ES}}$ et $\Rightarrow_{l,TPC_{ES}}$ à l'aide des lemmes (4.6) et (4.8).

Lemme 4.6 (Equivalence entre $\hookrightarrow$ et $\tilde{\hookrightarrow}$) Si M est un terme de TPC_{ES} et si L est un terme pur de TPC_{ES} alors $M \hookrightarrow L \Leftrightarrow M \tilde{\hookrightarrow}^* L$ et même si M n'est par pur $M \hookrightarrow L \Leftrightarrow M \tilde{\hookrightarrow}^+ L$.

<u>Preuve:</u>
<u>De $\Rightarrow$:</u> Par récurrence sur la taille de l'arbre de preuve de $M \hookrightarrow L$.
<u>Cas de base:</u> Si $M \hookrightarrow M$ alors M est pur et on a donc $M \tilde{\hookrightarrow}^* L$.
<u>Cas inductif:</u>

- $M \equiv x[\ x\ /\ M_1\ ]\ \Theta \hookrightarrow L$ provient de $M_1\Theta \hookrightarrow L$. Alors par hypothèse de récurrence $M_1\Theta\ \tilde{\hookrightarrow}^*\ L$ et donc $M \tilde{\hookrightarrow}^+ L$.

- $M \equiv y[\theta]\Theta \hookrightarrow L$ provient de $y\Theta \hookrightarrow L$ avec y$\neq$Dom(Θ). Alors par hypothèse de récurrence $y\Theta\ \tilde{\hookrightarrow}^*\ L$ et donc $M \tilde{\hookrightarrow}^+ L$.

- $M \equiv f(M_1, \ldots, M_n) \hookrightarrow f(K_1, \ldots, K_n)$ avec f $\in \{\texttt{let}\ \_\ \texttt{be}\ P\ \texttt{in}\ \_; \langle \_, \_ \rangle; \lambda$ $P.\ \_; \texttt{inr}(\_);\ \texttt{inl}(\_)\}$ et $\forall i,\ M_i \hookrightarrow K_i$. Deux cas sont possibles. Dans les deux cas $\forall i,\ M_i \tilde{\hookrightarrow}^* K_i$ par hypothèse de récurrence. Premier cas: Θ est vide alors l'un des M_i n'est pas pur donc $M\ \tilde{\hookrightarrow}^*\ f(K_1, M_2, \ldots, M_n)\ \tilde{\hookrightarrow}^*$ $\ldots\ \tilde{\hookrightarrow}^*\ f(K_1, \ldots, K_{n-1}, M_n)\ \tilde{\hookrightarrow}^*\ f(K_1, \ldots, K_n)$ et l'une des * est en fait un $^+$ puisque l'un des M_i n'est pas pur. Deuxième cas: Θ n'est pas vide alors $M \tilde{\hookrightarrow} f(M_1\Theta, \ldots, M_n\Theta)$ et l'on est revenu au cas précédent.

- $M \equiv (z\ \texttt{of}\ M_1\ \texttt{is}\ P\ \texttt{in}\ N)[\theta]\Theta \hookrightarrow L$ provient de $(z\ \ \texttt{of}\ M_1[\theta]\ \ \texttt{is}\ \ P\ \ \texttt{in}$ $N[\theta])\Theta\ \hookrightarrow\ L$ et z$\neq$ $Dom(\theta)$. Par hypothèse de récurrence $(z\ \ \texttt{of}\ M_1[\theta]$ $\texttt{is}\ \ P\ \ \texttt{in}\ \ N[\theta])\Theta\ \tilde{\hookrightarrow}^*\ L$ donc $M \tilde{\hookrightarrow}^+ L$.

- $M \equiv (z\ \texttt{of}\ M_1\ \texttt{is}\ P\ \texttt{in}\ N)[z/L]\Theta \hookrightarrow L$ provient de $(L\ \ \texttt{of}\ M_1\ \ \texttt{is}\ \ P\ \ \texttt{in}$ $N)\Theta\ \hookrightarrow\ K$. Par hypothèse de récurrence $(L\ \texttt{of}\ M_1\ \texttt{is}\ P\ \texttt{in}\ N)\Theta \tilde{\hookrightarrow}^* K$ donc $M \tilde{\hookrightarrow}^+ K$

- $M\ \equiv\ (\lambda P.J\ \texttt{of}\ M_1\ \texttt{is}\ Q\ \texttt{in}\ N)\Theta\ \hookrightarrow\ \lambda P.J_1\ \texttt{of}\ M_2\ \texttt{is}\ Q\ \texttt{in}\ N_1$ avec Θ non vide et $J\Theta \hookrightarrow J_1$, $M_1\Theta \hookrightarrow M_2$ et $N\Theta \hookrightarrow N_1$. Par hypothèse de récurrence $J\Theta \tilde{\hookrightarrow}^* J_1$, $M_1\Theta \tilde{\hookrightarrow}^* M_2$ et $N\Theta \tilde{\hookrightarrow}^* N_1$, donc $M\ \tilde{\hookrightarrow}\ \lambda\ P.\ J\Theta\ \ \texttt{of}\ M_1\Theta$ $\texttt{is}\ \ Q\ \ \texttt{in}\ \ N\Theta\ \tilde{\hookrightarrow}^*\ L$.

- $[M \mid_{\mathtt{T}} N][\theta]\Theta \hookrightarrow L$ provient de $[M[\theta] \mid_{\mathtt{T}} N[\theta]]\Theta \hookrightarrow L$ et $\mathtt{T} \neq Dom(\theta)$. Par hypothèse de récurrence $[M[\theta] \mid_{\mathtt{T}} N[\theta]]\Theta \tilde{\hookrightarrow}^* L$ donc on bien $M \tilde{\hookrightarrow}^+ L$.

- $M \equiv [M_1 \mid_\xi N][\xi/\mathtt{K}]\Theta \hookrightarrow [M_2 \mid_\mathtt{K} N_1]$ provient de $M_1[\xi/\mathtt{K}]\Theta \hookrightarrow M_2$ et $N[\xi/\mathtt{K}]\Theta \hookrightarrow N_1$. Alors par hypothèse de récurrence $M_1[\xi/\mathtt{K}]\Theta \tilde{\hookrightarrow}^* M_2$ et $N[\ \xi\ /\ \mathtt{K}\ ]\ \Theta \tilde{\hookrightarrow}^* N_1$. De plus $M \tilde{\hookrightarrow} [M_1[\xi/\mathtt{K}]\Theta \mid_\mathtt{K} N[\xi/\mathtt{K}]\Theta]$ donc $M \tilde{\hookrightarrow}^+ L$.

- $M \equiv [M_1 \mid_\mathtt{T} N] \hookrightarrow [M_2 \mid_\mathtt{T} N_1]$ provient de $M_1 \hookrightarrow M_2$ et $N \hookrightarrow N_1$ et M_1 ou N_1 n'est pas pur. Par hypothèse de récurrence $M_1 \tilde{\hookrightarrow}^* M_2$ et $N \tilde{\hookrightarrow}^* N_1$ et donc $M \tilde{\hookrightarrow}^+ [M_2 \mid_\mathtt{T} N_1]$.

<u>De $\Leftarrow$:</u> Par récurrence sur la longueur k de la dérivation de $M \tilde{\hookrightarrow}^* L$.

<u>Cas de base:</u> Si k=0 alors M est pur et donc $M \hookrightarrow M$
<u>Cas inductif:</u> Si k>0,

- $M \equiv x[x/M_1]\Theta \tilde{\hookrightarrow} M_1\Theta \tilde{\hookrightarrow}^* L$. Par hypothèse de récurrence $M_1\Theta \hookrightarrow L$ donc $M \hookrightarrow L$.

- $M \equiv y[\theta]\Theta \tilde{\hookrightarrow} y\Theta \tilde{\hookrightarrow}^* L$ avec y$\neq$ $Dom(\theta)$. Par hypothèse de récurrence $y\Theta \hookrightarrow L$ donc $M \hookrightarrow L$.

- $M \equiv f(M_1, \ldots, M_N)\Theta \tilde{\hookrightarrow} f(M_1\Theta, \ldots, M_n\Theta) \tilde{\hookrightarrow}^* L$ avec Θ non vide et f$\in$ $\{\mathtt{let}\ \_\ \mathtt{be}\ P\ \mathtt{in}\ \_; \langle \_, \_ \rangle; \lambda P.\_; \mathtt{inr}(\_); \mathtt{inl}(\_)\}$. Par hypothèse de récurrence $f(M_1\Theta, \ldots, M_n\Theta) \hookrightarrow L$ et donc $M \hookrightarrow L$.

- $M \equiv g(K_1, \ldots, K_p, M_{p+1}, \ldots, M_n) \tilde{\hookrightarrow}^* g(K_1, \ldots, K_n)$ avec g n'importe quel autre constructeur autre que la substitution. Trois cas sont possibles: g $\in$ $\{\mathtt{let}\ \_\ \mathtt{be}\ P\ \mathtt{in}\ \_; \langle \_, \_ \rangle; \lambda P.\_; \mathtt{inr}(\_); \mathtt{inl}(\_)\}$, g=$\_\ \mathtt{of}\ \_\ \mathtt{is}\ P\ \mathtt{in}\ \_$ et g=$[\_ \mid_\mathtt{T} \_]$. Dans les trois cas une règle de $\hookrightarrow$ s'applique pour que l'on obtienne $M \hookrightarrow L$.

- $(z\ \mathtt{of}\ M_1\ \mathtt{is}\ P\ \mathtt{in}\ N)[z/L_1]\Theta \tilde{\hookrightarrow} (L_1\ \mathtt{of}\ M_1\ \mathtt{is}\ P\ \mathtt{in}\ N)\Theta \tilde{\hookrightarrow}^* L$. Par hypothèse de récurrence $(L_1\ \mathtt{of}\ M_1\ \mathtt{is}\ P\ \mathtt{in}\ N)\Theta \hookrightarrow L$ donc $M \hookrightarrow L$.

- $(z\ \mathtt{of}\ M_1\ \mathtt{is}\ P\ \mathtt{in}\ N)[\theta]\Theta \tilde{\hookrightarrow} (z\ \mathtt{of}\ M_1[\theta]\ \mathtt{is}\ P\ \mathtt{in}\ N[\theta])\Theta \tilde{\hookrightarrow}^* L$ alors par hypothèse de récurrence $(z\ \mathtt{of}\ M_1[\theta]\ \mathtt{is}\ P\ \mathtt{in}\ N[\theta])\Theta \hookrightarrow L$ donc $M \hookrightarrow L$.

- Tous les autre cas se traitent de manière similaire.

$\square$

Remarque 4.7 On a en fait $\tilde{\rightsquigarrow} = \rightsquigarrow_l$. Nous utiliserons désormais indifféremment les deux notation.

Lemme 4.8 Si M est un terme clos de TPC_{ES} et si L est un fnp$^+$ alors $M \searrow L \Leftrightarrow M \nearrow^* L$

<u>Preuve:</u>
<u>De $\Rightarrow$:</u> Par récurrence sur la taille de l'arbre de preuve de $M \searrow L$.
<u>Cas de base:</u> Si $M \searrow M$ alors $M \nearrow^* M$ est évident.
<u>Cas inductif:</u>

- Si $M \equiv \lambda P.J\ \mathtt{of}\ M_1\ \mathtt{is}\ Q\ \mathtt{in}\ N \searrow L$ provient de $\mathtt{let}\ (\mathtt{let}\ M_1\ \mathtt{be}\ P\ \mathtt{in}\ J)\ \mathtt{be}\ Q\ \mathtt{in}\ N \searrow L$ alors, par hypothèse de récurrence, $\mathtt{let}\ (\mathtt{let}\ M_1\ \mathtt{be}\ P\ \mathtt{in}\ J)\ \mathtt{be}\ Q\ \mathtt{in}\ N \nearrow^* L$ et donc $M \nearrow^+ L$.

– Si $M \equiv$ $\mathtt{let}\ J\ \mathtt{be}\ \langle P_1, P_2\rangle\ \mathtt{in}\ N \searrow L$ avec $J \searrow \langle M_1, M_2\rangle$ et $\mathtt{let}\ M_1\ \mathtt{be}\ P_1\ \mathtt{in}\ (\mathtt{let}\ M_2\ \mathtt{be}\ P_2\ \mathtt{in}\ N) \searrow L$ alors hypothèse de récurrence $J \nearrow^* \langle M_1, M_2\rangle$ et $\mathtt{let}\ M_1\ \mathtt{be}\ P_1\ \mathtt{in}\ (\mathtt{let}\ M_2\ \mathtt{be}\ P_2\ \mathtt{in}\ N) \nearrow^* L$ donc $M \nearrow^*\ \mathtt{let}\ \langle M_1, M_2\rangle\ \mathtt{be}\ \langle P_1, P_2\rangle\ \mathtt{in}\ N \nearrow\ \mathtt{let}\ M_1\ \mathtt{be}\ P_1\ \mathtt{in}\ (\mathtt{let}\ M_2\ \mathtt{be}\ P_2\ \mathtt{in}\ N) \nearrow^* L$.

– Les autres cas sauf celui où M $\equiv M_1[\theta]\Theta$ se traitent de manière similaire à l'un des deux cas précédents.

– Si $M \equiv M_1[\theta]\Theta \searrow L$ avec $M_1[\theta] \rightsquigarrow_l M_2$ et $M_2\Theta \searrow L$ alors par hypothèse de récurrence $M_2\Theta \nearrow^* L$. De plus on a clairement $M_1[\theta] \nearrow M_2$ donc on a bien $M \nearrow^* L$.

De $\Leftarrow$: Par récurrence sur la longueur k de la dérivation de $M \nearrow^k L$.
Cas de base: Si k=0 alors M est un fnp$^+$ donc on a bien $M \searrow M$.
Cas inductif: Si k> 0,

– Si $M \equiv \lambda P.J\ \mathtt{of}\ 2\ \mathtt{is}\ M_1\ \mathtt{in}\ QN \nearrow\ \mathtt{let}\ (\mathtt{let}\ M_1\ \mathtt{be}\ P\ \mathtt{in}\ J)\ \mathtt{be}\ Q\ \mathtt{in}\ N \nearrow^* L$ alors par hypothèse de récurrence $\mathtt{let}\ (\mathtt{let}\ M_1\ \mathtt{be}\ P\ \mathtt{in}\ J)\ \mathtt{be}\ Q\ \mathtt{in}\ N \searrow L$ et donc $M \searrow L$.

– Si $M \equiv \mathtt{let}\ \langle M_1, M_2\rangle\ \mathtt{be}\ \langle P_1, P_2\rangle\ \mathtt{in}\ N \nearrow^* L$ on a le résultat par hypothèse de récurrence.

– Tous les autres cas, autres que ceux où une substitution est le constructeur de tête de M et celui où M $\equiv \mathtt{let}\ M_1\ \mathtt{be}\ P\ \mathtt{in}\ N$ et aucune règle de réduction du let ne s'applique, se traitent de manière similaire.

– Si M $\equiv \mathtt{let}\ M_1\ \mathtt{be}\ P\ \mathtt{in}\ N$ et aucune règle de réduction du let ne s'applique alors il existe une fnp$^+$ M_2 tel que $M_1 \nearrow^+ M_2$. Par hypothèse de récurrence on a alors $M_1 \searrow M_2$. On peut alors appliquer la règle correspondante de $\searrow$.

– Si $M \equiv M_1[\theta]\Theta \nearrow M_2\Theta \nearrow^* L$ avec $M_1[\theta] \tilde{\rightsquigarrow} M_2$ alors on remarque que $M_1[\theta] \rightsquigarrow_l M_2$ et a par hypothèse de récurrence $M_2\Theta \searrow L$ on a donc $M \searrow L$.

□

Théorème 4.9 Si M est un terme clos et pur de TPC_{ES} et si K est une fnp alors $M \Downarrow_{l,TPC_{ES}} K \Leftrightarrow M \Rightarrow^*_{l,TPC_{ES}} K$.

Preuve:
Par les lemmes (4.6) et (4.8).

□

4.3 Un évaluateur strict pour TPC

L'évaluateur $\Rightarrow_{e,TPC}$ implante la stratégie stricte pas à pas pour TPC. Nous allons montrer son équivalence avec $\Downarrow_{e,TPC}$. Nous montrerons aussi qu'il préserve les types et que tout terme clos et bien typé de TPC est soit une fns soit réductible par $\Rightarrow_{e,TPC}$. Ceci nous permettra alors de conclure que tout terme clos et bien typé de TPC est évaluable par $\Downarrow_{e,TPC}$.

Définition 4.3 La relation $\Rightarrow_{e,TPC}$ est une relation entre les termes clos de TPC.

Dans les règles qui suivent, les termes de la forme K ou K_i sont supposés être des fns.

$$\overline{\lambda P.J \texttt{ of } M \texttt{ is } Q \texttt{ in } N \Rightarrow_{e,TPC} \texttt{let } (\texttt{let } M \texttt{ be } P \texttt{ in } J) \texttt{ be } Q \texttt{ in } N}$$

$$\overline{\texttt{let } \langle K_1,K_2\rangle \texttt{ be } \langle P_1,P_2\rangle \texttt{ in } N \Rightarrow_{e,TPC} \texttt{let } K_1 \texttt{ be } P_1 \texttt{ in } (\texttt{let } K_2 \texttt{ be } P_2 \texttt{ in } N)}$$

$$\overline{\texttt{let } K \texttt{ be } @(P_1,P_2) \texttt{ in } N \Rightarrow_{e,TPC} \texttt{let } M \texttt{ be } P_1 \texttt{ in } (\texttt{let } K \texttt{ be } P_2 \texttt{ in } N)}$$

$$\overline{\texttt{let inl}(K) \texttt{ be } (P|_\xi Q) \texttt{ in } N \Rightarrow_{e,TPC} \texttt{let } K \texttt{ be } P \texttt{ in } N\{\xi \leftarrow \texttt{L}\}}$$

$$\overline{\texttt{let inr}(K) \texttt{ be } (P|_\xi Q) \texttt{ in } N \Rightarrow_{e,TPC} \texttt{let } K \texttt{ be } Q \texttt{ in } N\{\xi \leftarrow \texttt{R}\}}$$

$$\overline{\texttt{let } \lambda P.J \texttt{ be } \#z \texttt{ in } N \Rightarrow_{e,TPC} N\{z \leftarrow \lambda P.J\}}$$

$$\overline{\texttt{let } K \texttt{ be } x \texttt{ in } N \Rightarrow_{e,TPC} N\{x \leftarrow K\}}$$

$$\overline{\texttt{let } M \texttt{ be \_ in } N \Rightarrow_{e,TPC} N}$$

$$\overline{[M|_{\texttt{L}}N] \Rightarrow_{e,TPC} M}$$

$$\overline{[M|_{\texttt{R}}N] \Rightarrow_{e,TPC} N}$$

$$\frac{M \Rightarrow_{e,TPC} L}{\texttt{inl}(M) \Rightarrow_{e,TPC} \texttt{inl}(L)}$$

$$\frac{M \Rightarrow_{e,TPC} L}{inr(M) \Rightarrow_{e,TPC} \texttt{inr}(L)}$$

$$\frac{M \Rightarrow_{e,TPC} L}{\langle M,N\rangle \Rightarrow_{e,TPC} \langle L,N\rangle} \prec \text{Si M n'est pas une fns}$$

$$\frac{N \Rightarrow_{e,TPC} L}{\langle K,N\rangle \Rightarrow_{e,TPC} \langle K,L\rangle}$$

$$\frac{M \Rightarrow_{e,TPC} M_1}{\texttt{let } M \texttt{ be } P \texttt{ in } N \Rightarrow_{e,TPC} \texttt{let } M_1 \texttt{ be } P \texttt{ in } N} \prec \text{si on ne peut appliquer une des autres règles}$$

On voit beaucoup mieux ici que sur les évaluateurs d'un grand pas les différences entre la stratégie paresseuse et la stratégie stricte. En effet, alors que globalement les règles des relations $\Downarrow_{l,TPC}$ et $\Downarrow_{e,TPC}$ étaient très semblables, celles des relations $\Rightarrow_{l,TPC}$ et $\Rightarrow_{e,TPC}$ diffèrent plus. En particulier dans de nombreuses règles de $\Rightarrow_{e,TPC}$ on est obligé de préciser *explicitement* que le membre gauche contient une fns.

Remarque 4.10 On remarque que $\Rightarrow_{e,TPC} \subset \rightarrow_{tpc}$. En fait les relations $\Rightarrow_{e,TPC}$ et $\rightarrow_{tpc}$ se ressemblent beaucoup. Leur principale différence est que $\Rightarrow_{e,TPC}$ "déterminise" la relation $\rightarrow_{tpc}$.

Nous allons maintenant donner un exemple de réduction grâce au terme M défini comme suit: si $M_1 = \langle \lambda x.x, \texttt{let } \lambda y.y \texttt{ be } z \texttt{ in } z\rangle$ alors $M = \texttt{let } M_1 \texttt{ be } t \texttt{ in } \lambda u.t$.

$$\dfrac{\dfrac{\texttt{let}\ \lambda y.y\ \texttt{be}\ z\ \texttt{in}\ z \Rightarrow_{e,TPC} z\{z \leftarrow \lambda y.y\} \equiv \lambda y.y}{\langle \lambda x.x, \texttt{let}\ \lambda y.y\ \texttt{be}\ z\ \texttt{in}\ z\rangle \Rightarrow_{e,TPC} \langle \lambda x.x, \lambda y.y\rangle \equiv M_3}}{M \Rightarrow_{e,TPC} \texttt{let}\ M_3\ \texttt{be}\ t\ \texttt{in}\ \lambda u.t \equiv M_2} \qquad \dfrac{}{M_2 \Rightarrow_{e,TPC} (\lambda u.t)\{t \leftarrow M_3\} \equiv \lambda u.M_3}$$

Les deux propositions suivantes montrent l'équivalence entre $\Rightarrow_{e,TPC}$ et $\Downarrow_{e,TPC}$.

Proposition 4.11 (Equivalence entre $\Downarrow_{e,TPC}$et $\Rightarrow_{e,TPC}$ (I)) Si M est u terme clos de TPC et si $M \Downarrow_{e,TPC} K$ alors $M \Rightarrow^*_{e,TPC} K$ et même $M \Rightarrow^+_{e,TPC} K$ si M n'est pas une fnp .

<u>Preuve:</u>
Par récurrence sur la taille de l'arbre de preuve de M $\Downarrow_e$ K.

□

Proposition 4.12 (Equivalence entre $\Downarrow_{e,TPC}$ et $\Rightarrow_{e,TPC}$ (II)) Si M est un terme clos de TPC et si $M \Rightarrow^*_{e,TPC} K$ et K est une fns alors $M \Downarrow_{e,TPC} K$.

<u>Preuve:</u>
Par récurrence sur la longueur de la dérivation de M à K par $\Rightarrow_{e,TPC}$.

□

Le lemme suivant montre que tout terme de TPC clos et bien typé est soit une fns donc un résultat pour $\Downarrow_{e,TPC}$ soit réductible par $\Rightarrow_{e,TPC}$.

Lemme 4.13 Si M est un programme de TPC et si M n'est pas une fns alors $M \Rightarrow_{e,TPC} N$ pour un certain N.

<u>Preuve:</u>
Par induction sur la structure de M.

□

On tire en particulier de la remarque (4.10) que, comme $\rightarrow_{tpc}$, $\Rightarrow_{l,TPC}$ a la propriété de préserver les types.

Le théorème suivant montre que tout terme clos et bien typé de TPC est évaluable par $\Downarrow_e$

Théorème 4.14 Si M est un programme de TPC alors $M \Downarrow_{e,TPC} K$ pour un certain K.

<u>Preuve:</u>
Par le lemme (4.13) $M \Rightarrow^*_{e,TPC} K$ pour un certain K avec K une fns et par la proposition (4.3) $M \Downarrow_e K$.

□

4.4 Un évaluateur strict pour TPC_{ES}

L'évaluateur $\Rightarrow_{e,TPC_{ES}}$ implante la stratégie stricte pas à pas pour TPC_{ES}. Il est équivalent à $\Downarrow_{e,TPC_{ES}}$.

Définition 4.4 Nous nous resservons ici de la relation $\tilde{\hookrightarrow}$ comme relation de purification. La relation $\Rightarrow_{e,TPC_{ES}}$ est quand à elle une relation sur les terme clos de TPC_{ES}.

$$\frac{M \tilde{\hookrightarrow} M_1}{M \Rightarrow_{e,TPC_{ES}} M_1} \prec \text{Si } M \text{ n'est pas pur}$$

Dans la suite de la description des règles de $\Rightarrow_{e,TPC_{ES}}$ les membres de gauches des règles sont purs et les termes notés K où K_i sont des fns.

$$\frac{}{\lambda P.J \ \mathtt{of}\ M \ \mathtt{is}\ Q \ \mathtt{in}\ N \Rightarrow_{e,TPC_{ES}} \mathtt{let}\ (\mathtt{let}\ M \ \mathtt{be}\ P \ \mathtt{in}\ J) \ \mathtt{be}\ Q \ \mathtt{in}\ N}$$

$$\frac{}{\mathtt{let}\ \langle K_1,K_2\rangle \ \mathtt{be}\ \langle P_1,P_2\rangle \ \mathtt{in}\ N \Rightarrow_{e,TPC_{ES}} \mathtt{let}\ K_1 \ \mathtt{be}\ P_1 \ \mathtt{in}\ (\mathtt{let}\ K_2 \ \mathtt{be}\ P_2 \ \mathtt{in}\ N)}$$

$$\frac{}{\mathtt{let}\ K \ \mathtt{be}\ @(P_1,P_2) \ \mathtt{in}\ N \Rightarrow_{e,TPC_{ES}} \mathtt{let}\ M \ \mathtt{be}\ P_1 \ \mathtt{in}\ (\mathtt{let}\ M \ \mathtt{be}\ P_2 \ \mathtt{in}\ N)}$$

$$\frac{}{\mathtt{let}\ \mathtt{inl}(K) \ \mathtt{be}\ (P|_\xi Q) \ \mathtt{in}\ N \Rightarrow_{e,TPC_{ES}} \mathtt{let}\ K \ \mathtt{be}\ P \ \mathtt{in}\ N[\xi/\mathtt{L}]}$$

$$\frac{}{\mathtt{let}\ \mathtt{inr}(K) \ \mathtt{be}\ (P|_\xi Q) \ \mathtt{in}\ N \Rightarrow_{e,TPC_{ES}} \mathtt{let}\ K \ \mathtt{be}\ Q \ \mathtt{in}\ N[\xi/\mathtt{R}]}$$

$$\frac{}{\mathtt{let}\ \lambda P.M \ \mathtt{be}\ \#z \ \mathtt{in}\ N \Rightarrow_{e,TPC_{ES}} N[z/\lambda P.M]}$$

$$\frac{}{\mathtt{let}\ K \ \mathtt{be}\ x \ \mathtt{in}\ N \Rightarrow_{e,TPC_{ES}} N[x/K]}$$

$$\frac{}{\mathtt{let}\ M \ \mathtt{be}\ \_ \ \mathtt{in}\ N \Rightarrow_{e,TPC_{ES}} N}$$

$$\frac{M \Rightarrow_{e,TPC_{ES}} M_1}{\mathtt{let}\ M \ \mathtt{be}\ P \ \mathtt{in}\ N \Rightarrow_{e,TPC_{ES}} \mathtt{let}\ M_1 \ \mathtt{be}\ P \ \mathtt{in}\ N} \prec \text{Si aucune des règles précédentes ne s'applique}$$

$$\frac{M \Rightarrow_{e,TPC_{ES}} M_1}{\mathtt{inl}(M) \Rightarrow_{e,TPC_{ES}} \mathtt{inl}(M_1)} \prec \text{Si M n'est pas une fns}$$

$$\frac{M \Rightarrow_{e,TPC_{ES}} M_1}{\mathtt{inr}(M) \Rightarrow_{e,TPC_{ES}} \mathtt{inr}(M_1)} \prec \text{Si M n'est pas une fns}$$

$$\frac{M \Rightarrow_{e,TPC_{ES}} M_1}{\langle M,N\rangle \Rightarrow_{e,TPC_{ES}} \langle M_1,N\rangle} \prec \text{Si M n'est pas une fns}$$

$$\frac{N \Rightarrow_{e,TPC_{ES}} N_1}{\langle K,N\rangle \Rightarrow_{e,TPC_{ES}} \langle K,N_1\rangle} \prec \text{Si N n'est pas une fns}$$

Nous allons maintenant donner la réduction du terme M défini comme suit: si $M_1 = \langle \lambda x.x, \mathtt{let}\ \lambda y.y \ \mathtt{be}\ z \ \mathtt{in}\ z\rangle$ alors $M = \mathtt{let}\ M_1 \ \mathtt{be}\ t \ \mathtt{in}\ \lambda u.t$.

$$\frac{\dfrac{\mathtt{let}\ \lambda y.y \ \mathtt{be}\ z \ \mathtt{in}\ z \Rightarrow_{e,TPC_{ES}} z[z/\lambda y.y] \equiv M_2}{\langle \lambda x.x, \mathtt{let}\ \lambda y.y \ \mathtt{be}\ z \ \mathtt{in}\ z\rangle \Rightarrow_{e,TPC_{ES}} \langle \lambda x.x, M_2\rangle}}{M \Rightarrow_{e,TPC_{ES}} \mathtt{let}\ \langle \lambda x.x, M_2\rangle \ \mathtt{be}\ t \ \mathtt{in}\ \lambda u.t \equiv M_3} \qquad \frac{\dfrac{M_2 \tilde{\hookrightarrow} \lambda y.y}{M_3 \tilde{\hookrightarrow} \langle \lambda x.x, \lambda y.y\rangle \equiv M_4}}{M_3 \Rightarrow_{e,TPC_{ES}} \mathtt{let}\ M_4 \ \mathtt{be}\ t \ \mathtt{in}\ \lambda u.t \equiv M_5}$$

$$\frac{}{M_5 \Rightarrow_{e,TPC_{ES}} (\lambda u.t)[t/M_4]} \qquad \frac{(\lambda u.t)[t/M_4] \tilde{\hookrightarrow} \lambda u.t[t/M_4]}{(\lambda u.t)[t/M_4] \Rightarrow_{e,TPC_{ES}} \lambda u.t[t/M_4]} \qquad \frac{\lambda u.t[t/M_4] \tilde{\hookrightarrow} \lambda t.M_4}{\lambda u.t[t/M_4] \Rightarrow_{e,TPC_{ES}} \lambda t.M_4}$$

Le théorème suivant montre l'équivalence entre $\Downarrow_{e,TPC_{ES}}$ et $\Rightarrow_{e,TPC_{ES}}$.

Théorème 4.15 Si M est clos et si K est une fnp alors $M \Downarrow_{e,TPC_{ES}} K \Leftrightarrow M \Rightarrow^*_{e,TPC_{ES}} K$.

Preuve:
Par induction sur la taille de l'arbre de preuve de $M \Downarrow_e K$ pour $\Rightarrow$ et sur la longueur de la dérivation de $M \Rightarrow^*_{e,TPC_{ES}} K$ pour $\Leftarrow$.

□

4.5 Des résultats pour TPC_{ES}

On récupère alors le fait que $\Downarrow_{l,TPC_{ES}}$ et $\Downarrow_{e,TPC_{ES}}$ évaluent tous les termes clos et bien typés de TPC_{ES} comme le montrent les deux théorèmes suivant.

Théorème 4.16 Si M est un programme de TPC_{ES} alors $M \Downarrow_{l,TPC_{ES}} K$ pour un certain K.

Preuve:
Par application du théorème (4.5) il existe un K tel que $ES(M) \Downarrow_{l,TPC} K$ donc par le théorème (3.13) $ES(M) \Downarrow_{l,TPC_{ES}} K$. Par la proposition (3.11) on a M $\Downarrow_{l,TPC_{ES}}$ K.

□

Théorème 4.17 Si M est un programme de TPC_{ES} alors $M \Downarrow_{e,TPC_{ES}} K$ pour un certain K.

Preuve:
On obtient ce résultat comme une application immédiate du théorème (4.14) et du lemme (3.24).

5 Conclusions et développements possibles de ce travail

Nous avons durant ce stage défini les stratégies paresseuse et stricte pour les calculs TPC et TPC_{ES} introduits par Delia Kesner et Serenella Cerrito [4]. Nous avons montré les propriétés vérifiées par ces évaluateurs à savoir leur déterminisme, leur correction par rapport à TPC ou TPC_{ES} suivant les cas, la préservation des types et les différents types d'équivalences entres des évaluateurs correspondant.

L'un des "problèmes" existant au moment de l'implantation d'un langage est celui de l'α-conversion c'est à dire du renommage automatique des variables. Une manière de résoudre ce problème est de changer les noms des variables en notations de Bruijn [8, 9]. Ce passage est l'une des futures extensions de TPC et TPC_{ES} envisageables.

Une autre extension possible de ce travail serait donner un développement polymorphe (pour les types, les motifs et les termes) de TPC et TPC_{ES} par exemple grâce à une méthode similaire à celle proposé dans [7].

D'autres problèmes se posent avec TPC et TPC_{ES}. Il serait intéressant par exemple d'inclure dans ces calculs les types inductifs comme les listes ou les arbres.

Un problème purement technique est la règle $Dapp_1$ de TPC_{ES} qui nous a obligé tout au long de ce rapport à traiter à part les termes de la forme z `of` M `is` P `in` N. Cette règle est rendue nécessaire par le système de typage actuel de TPC_{ES}. Il nous semblerait intéressant de modifier ce système de typage de manière à pouvoir traiter les termes de la forme z `of` M `is` P `in` N comme tous les autres.

Enfin, les problèmes de la vérification et de l'inférence de type ne sont pas à l'heure actuelle traités. Ces problèmes étant vitaux lors de la définition d'un langage de programmation typé, il sera nécessaire de les traiter.

Remerciements

Avant de conclure définitivement ce rapport, je tiens à remercier Serenella Cerrito qui m'a fait part de ces très intéressantes observations sur ce rapport et Delia Kesner qui m'a guidé tout au long de ce stage.

Références

[1] M. Abadi, L. Cardelli, P-L. Curien, and J-J. Lévy. Explicit substitutions. *Journal of Functional Programming*, 4(1):375–416, 1991.

[2] R. Bloo. *Preservation of Termination for Explicit Substitution*. PhD thesis, Eindhoven University of Technology, 1997.

[3] R. Bloo and K. Rose. Combinatory reduction systems with explicit substitution that preserve strong normalisation. In *Proc. 7th Rewriting Techniques and Applications*, LNCS 1103, 1996.

[4] Serenella Cerrito and Delia Kesner. Pattern matching as cut elimination. In *Proc of the Fourteenth Annual IEEE Symposium on Logic in Computer Science*, 1999.

[5] A. Church. A note on the *entscheidungsproblem. ournal of Symbolic Logic*, 1:40–41, 101–102, 1936.

[6] A. Church. *The Calculi of Lambda-Conversion*, volume 6 of *Annals of Mathematical Studies*. Princeton University Press, Princeton, 1951.

[7] P. Courtieu. *Extension au second ordre du λ-calcul typé avec motifs*. Mémoire de DEA, Université de Paris XI, 1996.

[8] N. G. de Bruijn. Lambda calculus with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Proc. of the Koninklijke Nederlands Akademie*, 75(5):380–392, 1972.

[9] N. G. de Bruijn. Lambda calculus with namefree formulas involving symbols that represent reference transforming mappings. *Proc. of the Koninklijke Nederlands Akademie*, 81(3):1–9, September 1978. Dedicated to A. Heyting on the occasion of his 80th birthday on May 9,1978.

[10] H. Herbelin. A λ-calculus structure isomorphic to sequent calculus structure. In *Proc. 8th Conf. Computer Science Logic*, LNCS 933, 1994.

[11] P. Hudak, S. Peyton-Jones, and P. Wadler (editors). Report on the programming language haskell, a non-strict, purely functional language (version 1.2). Sigplan Notices, 1992.

[12] D. Kesner. Confluence of extensional and non-extensional lambda-calculi with explicit substitutions. *Theoretical Computer Science*. To appear.

[13] D. Kesner, L. Puel, and V. Tannen. A Typed Pattern Calculus. *Information and Computation*, 124(1):32–61, 1996.

[14] X. Leroy et al. The Caml Light system, release 0.6. Available on `ftp.inria.fr`, 1993.

[15] R. Milner, M. Tofte, and R. Harper. *The definition of Standard ML*. MIT Press, 1990.

[16] D. A. Turner. Miranda: A non-strict functional language with polymorphic types. In *Proc. IFIP International Conference on Functional Programming Languages and Computer Architecture*, LNCS 201, 1985.