RAIRO. INFORMATIQUE THÉORIQUE

B. COURCELLE
F. LAVANDIER

Définitions récursives par cas

RAIRO. Informatique théorique, tome 18, n° 2 (1984), p. 91-129
<http://www.numdam.org/item?id=ITA_1984__18_2_91_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 18, n° 2, 1983, p. 91 à 129)

DÉFINITIONS RÉCURSIVES PAR CAS (*)

par B. COURCELLE et F. LAVANDIER (1)

Résumé. — *Ce travail est le début d'une étude systématique de définitions récursives sans conditionnelles explicites qui s'introduisent naturellement dans de nombreuses situations : sémantique dénotationnelle, grammaires avec attributs, algorithmes sur les grammaires et d'autres encore.*

Abstract. — *This work begins a systematic study of certain recursive definitions which do not use explicitely the* **if**. . . **then**. . . **else** *construct. Such definitions naturally arise in denotational semantics, attribute grammars, algorithms on grammars and in other situations.*

0. INTRODUCTION

L'objet de ce travail est l'étude des *définitions récursives par cas* dont l'exemple de base est la définition suivante de la fonction factorielle :

$$\left.\begin{aligned} \mathrm{FAC}(0) &= 1, \\ \mathrm{FAC}(n+1) &= (n+1) \,.\, \mathrm{FAC}(n), \end{aligned}\right\} \tag{1}$$

que l'on peut aussi définir au moyen de l'unique équation :

$$\mathrm{FAC}(n) = \textbf{si } n=0 \textbf{ alors } 1 \textbf{ sinon } n \,.\, \mathrm{FAC}(n-1). \tag{2}$$

La différence entre ces deux définitions est très importante si l'on utilise des *schémas de programmes* pour les étudier.

(*) Reçu en septembre 1981, révisé en septembre 1983.

(1) Université de Bordeaux-I, U.E.R. de Mathématiques, 351, cours de la Libération, 33405 Talence Cedex.

Note : Ce travail a été effectué au titre du contrat d'A.T.P. n° 4275 du C.N.R.S. Une version préliminaire, en anglais, a été présentée au 8e colloque sur les Arbres en Algèbre et en Programmation, l'Aquila, Italie.

Note: *This work has been supported by the A.T.P. contract* 4275 *of C.N.R.S. A preliminary version, written in English, has been presented to the* 8*th C.A.A.P. (Colloquium on Trees in Algebra and Programming), L'Aquila, Italy (March* 1983).

0399-0540/84/02 91 39/$ 5.90/

Les définitions récursives du type (2) ont été étudiées au moyen de schémas de programmes récursifs polyadiques [3, 8, 13, 22]. Dans ces schémas le **si**. . . **alors**. . . **sinon** est considéré comme une fonction de base, au même titre que l'addition et la multiplication et non comme une structure de contrôle.

On associe à une équation telle que (2) un arbre infini dit *algébrique* (*voir* Courcelle [3, 4]) qui représente la sémantique de tous les programmes correspondant à toutes les interprétations possibles.

Cela veut dire que n'importe quelle fonction à trois arguments peut être prise comme interprétation du **si**. . . **alors**. . . **sinon** ce qui fait perdre ses propriétés particulières.

On peut les réintroduire au moyen de *restrictions* sur les interprétations que l'on associera au schéma. Et le seul moyen utilisable de faire cela semble être d'utiliser des *équations* telles que :

$$\textbf{si } x \textbf{ alors } (\textbf{si } x \textbf{ alors } y \textbf{ sinon } z) \textbf{ sinon } u = \textbf{si } x \textbf{ alors } y \textbf{ sinon } u. \qquad (3)$$

Mais ces équations (*cf.* [1]) introduisent des difficultés considérables et l'on atteint là les limites intrinsèques de cette approche (Courcelle [3]).

Considérer **si - alors - sinon** comme un élément de structure de contrôle conduit aux classes de schémas de programmes étudiées par Garland et Luckham [11] et à leur suite par Walker et Strong [27], Friedman [10] entre autres. Ces classes de schémas définissent en fait des *transducteurs d'arbres* particulièrement complexes.

Les schémas récursifs par cas que nous allons introduire seront plus proches des schémas de Walker et Strong [27] que des schémas récursifs polyadiques de Nivat [22] mais l'esprit dans lequel nous allons les étudier est radicalement différent.

Nous n'allons pas concentrer notre attention sur les résultats de décidabilité mais plutôt sur des caractérisations semi-décidables pratiquement utilisables (concernant la terminaison, la totalité, le déterminisme, etc. propriétés connues pour être indécidables) et les « familles de programmes analogues » associées aux schémas de programmes.

Les caractéristiques essentielles de notre approche sont les suivantes :

— Le **si - alors - sinon** est éliminé au moyen de la division d'une équation telle que (2) en plusieurs *clauses* (d'une façon inspirée du langage PROLOG). A notre sens, cela donne des définitions plus claires (elles sont utilisées en particulier pour décrire des algorithmes complexes dans Courcelle [5]).

— Plusieurs clauses peuvent être simultanément applicables, donnant ainsi une possibilité de non-déterminisme (contrairement au cas de PROLOG,

nous n'allons pas privilégier la première clause applicable, autrement dit l'ordre des clauses d'un programme sera sans effet sur sa sémantique). Nous étudierons plus spécialement les programmes *cohérents* c'est-à-dire ceux où le non-déterminisme du calcul n'empêche pas l'unicité du résultat.

— Nous utiliserons la théorie bien établie des systèmes de réécriture d'arbres (de termes) mais, contrairement à ce qui est fait dans [17] et [23] nous les utiliserons pour étudier des *schémas de programmes* et non des *programmes*.

Enfin l'aspect le plus original de ce travail est l'introduction de la notion de *semi-interprétation* dans laquelle seuls les membres gauches des clauses sont interprétés.

Mais cela ne s'applique qu'à certains schémas (appelés *spéciaux*). Et pour ces schémas, un bon nombre de propriétés intéressantes (terminaison, déterminisme, cohérence, etc.) ne dépendent que (de la syntaxe) des membres gauches des clauses et des semi-interprétations leur donnant un sens.

Nous allons maintenant donner deux exemples pour donner un sens intuitif à la distinction entre schémas spéciaux et non spéciaux.

0.1. Exemple : Fonction de transition d'un automate

Soit $A=\{a_1, a_2, \ldots, a_n\}$ un alphabet, Q un ensemble d'états, $h_i: Q \to Q$ une fonction ($i=1, \ldots, n$) qui est la fonction de transition associée à la lettre a_i.

La fonction de transition $\varphi: A^+ \times Q \to Q$ peut être définie ainsi :

$$\left.\begin{aligned} \varphi(a_i, q) &= h_i(q), \quad \text{pour} \quad i=1, \ldots, n, \\ \varphi(u.v, q) &= \varphi(v, \varphi(u, q)), \end{aligned}\right\} \tag{4}$$

où $u, v \in A^+$, $q \in Q$.

Noter au passage que cette définition est non-déterministe (car un mot de longueur supérieure à 2 peut être écrit $u.v$ de plusieurs façons différentes) mais cohérente car $\varphi(u, q)$ n'a qu'une valeur.

Remarquer également que le calcul de $\varphi(u, q)$ se termine pour tout u dans A^+ et tout q dans Q, *quelque soient les fonctions* $h_i: Q \to Q$.

De plus tout calcul (tel que $\varphi(a_2 a_2 a_3, q) \overset{*}{\to} q'$) peut être « remonté » en un calcul dans une interprétation partiellement symbolique (sa *trace*) où les fonctions h_i ne sont pas évaluées :

$$\varphi(a_2 a_2 a_3, q) \to \varphi(a_3, \varphi(a_2 a_2, q)) \to \varphi(a_3, \varphi(a_2, \varphi(a_2, q)))$$
$$\to \varphi(a_3, \varphi(a_2, h_2(q))) \to \varphi(a_3, h_2 h_2(q)) \to h_3 h_2 h_2(q).$$

En gros, pour deux interprétations de (4) ne différant que par les fonctions h_i, il existe une correspondance bijective entre les calculs d'un même terme $\varphi(u, q)$ pour les deux interprétations.

La raison en est que la suite des appels récursifs nécessaires au calcul de $\varphi(u, q)$ ne dépend que de u et ni de q, ni d'aucune valeur $h_i(q')$, $i=1, \ldots, n$, $q' \in Q$.

Ceci est typique d'un schéma spécial (et ne se produira pas dans l'exemple non spécial qui suit).

Noter enfin que la terminaison dépend de manière cruciale de A^+ et .; l'on n'a plus la terminaison si A^+ est remplacé par A^ω. □

0.2. Exemple : Fonction d'Ackermann

Soit maintenant le schéma récursif par cas qui suit :

$$\left.\begin{aligned} \psi(a, y) &= p(y, c), \\ \psi(p(x, b), a) &= \psi(x, d), \\ \psi(p(x, b), p(y, b)) &= \psi(x, \psi(p(x, e), y)). \end{aligned}\right\} \tag{5}$$

Considérons l'interprétation I_1 suivante :

$x, y \in \mathbb{N}$;

p représente l'addition;

a représente 0;

b, c, d, e représentent 1.

L'on obtient la définition classique de la fonction d'Ackermann. Rappelons que cette définition la définit comme totale.

Si maintenant on décide que c représente 0, tout le reste étant comme dans I_1, on obtient une autre interprétation I_2.

Le programme associé à I_2 est total également, mais les calculs d'un même terme tel que $\psi(2, 1)$ ne sont pas isomorphes. En particulier :

$\psi(2, 1) \xrightarrow{*}$ 5 dans I_1 en 14 étapes et :

$\psi(2, 1) \xrightarrow{*}$ 1 dans I_2 en 10 étapes (au moyen de la même stratégie de calcul). Plus précisément, les 7 premières étapes des deux calculs (en appel par valeur) utilisent les mêmes clauses dans les deux interprétations :

dans I_1	dans I_2
$\psi(2, 1) \xrightarrow{*} \psi(1, \psi(0, \psi(0, 1)))$	$\psi(2, 1) \xrightarrow{*} \psi(1, \psi(0, \psi(0, 1)))$
$\to \psi(1, \psi(0, 2))$	$\to \psi(1, \psi(0, 1))$
$\to \psi(1, 3)$	$\to \psi(1, 1)$
$\to \psi(0, \psi(1, 2))$	$\to \psi(0, \psi(1, 0))$

Mais les calculs se poursuivent de façons différentes car $\psi(1, 2)$ et $\psi(1, 0)$ s'évaluent (respectivement dans I_1 et I_2) par des clauses différentes.

Il ne peut donc pas y avoir « isomorphisme » entre les calculs de $\psi(2, 1)$ dans I_1 et dans I_2 et ceci est dû au fait que les suites d'appels récursifs sont controlées par les *valeurs* de sous expressions (non réduites à des variables) des membres droits des clauses du schéma.

Ainsi (5) est un exemple de schéma récursif par cas pour lequel on ne peut pas définir la trace d'un calcul.

Si maintenant I_3 est obtenue à partir de I_1 en donnant à e la valeur 2 tout le reste étant inchangé, on obtient dans I_3 le calcul divergent suivant :

$$\psi(1, 1) \to \psi(0, \psi(2, 0) \to \psi(0, \psi(1, 1)) \to \ldots$$

Ainsi ψ_{I_3} est une fonction partielle alors que ψ_{I_1} était totale. Cela ne se produirait pas si ψ était spécial. □

INTRODUCTION

Our purpose is the study of *by case recursive definitions* an example of which is the following definition of the factorial function:

$$\begin{aligned} &\mathrm{FAC}(0) = 1, \\ &\mathrm{FAC}(n+1) = (n+1) \,.\, \mathrm{FAC}(n). \end{aligned} \tag{1}$$

This function is often defined by the only equation:

$$\mathrm{FAC}(n) = \textbf{if } n=0 \textbf{ then } 1 \textbf{ else } n \,.\, \mathrm{FAC}(n-1). \tag{2}$$

The difference between these two systems is very important if we use *program schemes* for studying such definitions. Recursive definitions such as (2) have been studied by means of polyadic recursive schemes (*cf.* [3, 8, 13, 22]). In these schemes, the **if**. . . **then**. . . **else** construct is considered as a base function. With an equation such as (2) is associated infinite tree said *algebraic* (Courcelle [3, 4]) which represents the *semantics* of the corresponding program. Since an arbitrary tri-adic function can be taken as the meaning of **if**. . . **then**. . . **else**. . ., its specific properties are lost. They can be reintroduced by means of restrictions on the allowed interpretations. And the only manageable way to do so seems to be by equations like:

$$\textbf{if } x \textbf{ then } (\textbf{if } x \textbf{ then } y \textbf{ else } z) \textbf{ else } u = \textbf{if } x \textbf{ then } y \textbf{ else } u. \tag{3}$$

But even in this way things are difficult and the intrinsic limits of this approach are reached there (*see* Courcelle [3]).

Considering **if**. . . **then**. . . **else** as a piece of control structure leads to the classes of program schemes studied by Garland and Luckham [11], Walker and Strong [27], Friedman [10]. These works result in classes of *tree transducers* which are hard to study. The *by-case recursive schemes* we shall introduce will be closer to the latter than to the former ones but the spirit of our approach is different: we shall not focus our attention on (un) decidability results but rather on semi-decidable usable criteria (for termination, totality, determinism, etc.), and on program schemes as defining families of "similar" programs.

The main features of our approach are the following ones: **if**. . . **then**. . . **else** is avoided by a splitting of a recursive definition [like (2)] into several *clauses* (like in PROLOG). In our opinion, this gives more clear definitions (especially for complex algorithms as those of Courcelle [5]).

— Several clauses may be simultaneously applicable, yielding the possibility of non-determinism. We shall focus our attention on *coherent programs*, i. e. possibly non deterministic ones where every term has at most one value.

— We shall rest on the well-developped theory of tree-rewriting systems but we shall still deal with *program schemes* and not with *programs* as in [17] or [23].

— Finally, the major novelty is the introduction of the concept of a *semi-interpretation* in which only the left-hand sides of clauses are interpreted. For certain by case program schemes (termed *special*) many interesting properties like termination, determinism, coherence, etc. only depend on the left parts of the clauses and the semi-interpretation (i.e. the part of the interpretation) giving meaning to them. Let us precise this by two examples, one of which is "special".

Example 1: *Transition function of an automaton*

Let $A=\{a_1, \ldots, a_n\}$ be the alphabet, Q be the set of states, $h_i: Q \to Q$ be the transition function: $Q \to Q$ associated with letter a_i, $i=1, \ldots, n$. The transition function $\varphi: A^+ \times Q \to Q$ can be defined as follows:

$$\left.\begin{aligned} \varphi(a_i, q) &= h_i(q) \quad \text{for} \quad i=1, \ldots, n, \\ \varphi(u.v, q) &= \varphi(v, \varphi(u, q)). \end{aligned}\right\} \tag{4}$$

where u, v range over A^+ and q over Q. Note by passing that this by-case definition is nondeterministic (since a word of A^+ can be written as $u.v$ in

several different ways) but coherent [$\varphi(u, q)$ has a unique value]. The computation of $\varphi(u, q)$ terminates for all u in A^+ and q in Q whatever functions: $Q \to Q$ the h_i's might be. Moreover any computation (like $\varphi(a_2\, a_2\, a_3, q) \overset{*}{\to} q'$) can be "lifted" into a computation in a partially symbolic interpretation where the h_i's are unevaluated (we shall call it its *trace*):

$$\varphi(a_2\, a_2\, a_3, q) \to \varphi(a_3, \varphi(a_2\, a_2, q)) \to \varphi(a_3, \varphi(a_2, \varphi(a_2, q)))$$
$$\to \varphi(a_3, \varphi(a_2, h_2(q))) \to \varphi(a_3, h_2\, h_2(q)) \to h_3\, h_2\, h_2(q).$$

Roughly speaking, in any two interpretations of the scheme (4) differing only by the functions h_i's there is a bijective correspondence between the computations of any term $\varphi(u, q)$ (with u in A^+, q in Q).

The reason for this is that the sequence of recursive calls needed to compute $\varphi(u, q)$ only depends on u and not on any value $h_i(\ldots)$. This is typical of a special scheme. But note that the termination crucially depends on A^+ and .; it does not hold for A^ω instead of A^+.

Example 2: Ackermann's function

Let us now consider the by-case program scheme:

$$\begin{aligned} \psi(a, y) &= p(y, c), \\ \psi(p(x, b), a) &= \psi(x, d), \\ \psi(p(x, b), p(y, b)) &= \psi(x, \psi(p(x, e), y)), \end{aligned} \tag{5}$$

Let us define interpretation I_1 by letting x, y range over $\mathbf{N}$, p denote the addition, a denote 0, b, c, d, e all denote 1. We get the classical definition of Ackermann's function, known to be total.

Let now I_2 be such that c denotes 0, the other symbols being interpreted exactly as before; the associated program is still total, but the computations are not "similar"; for example $\psi(2, 1) \overset{*}{\to} 5$ in I_1 with 14 steps and $\psi(2, 1) \overset{*}{\to} 1$ in I_2 with 10 steps. More precisely, the 7 first steps of the computations of $\psi(2, 1)$ (with call-by-value) use the same clauses in both interpretations:

in I_1	in I_2
$\psi(2, 1) \overset{*}{\to} \psi(1, \psi(0, \psi(0, 1)))$	$\psi(2, 1) \overset{*}{\to} \psi(1, \psi(0, \psi(0, 1)))$
$\to \psi(1, \psi(0, 2))$	$\to \psi(1, \psi(0, 1))$
$\to \psi(1, 3)$	$\to \psi(1, 1)$
$\to \psi(0, \psi(1, 2))$	$\to \psi(0, \psi(1, 0))$

But the computations proceed in different directions since $\psi(1, 2)$ and $\psi(1, 0)$ are evaluated in two different ways.

This is due to the fact that the sequence of recursive calls is controlled by values computed with some functions appearing in the right hand-sides of clauses.

Hence (5) is an example of a by-case scheme for which the trace of a computation cannot be defined.

In interpretation I_3, derived from I_1 by the single change of the value of e to 2, we get a divergence:

$$\psi(1, 1) \to \psi(0, \psi(2, 0)) \to \psi(0, \psi(1, 1)) \to \ldots$$

The class of special schemes for which the concepts of semi-interpretation and trace make sense will be defined with help of sorts in a rather simple way in section 4.

1. DEFINITIONS

On rappelle rapidement les définitions classiques des notions de signature, de magma hétérogène, d'arbre puis on introduit les définitions récursives par cas et les schémas de programme correspondants. La terminologie et les notions sont celles de [6].

Divers exemples seront examinés dans la section 2.

1.1. Signatures

Soit $\mathscr{S}$ un ensemble de *sortes*. Une *$\mathscr{S}$-signature* (ou une *signature* si l'on sous-entend $\mathscr{S}$) est un ensemble F de symboles de fonctions, chacun d'eux, f, étant muni :

- d'une *arité* $\alpha(f)$ dans $\mathscr{S}^*$.
- d'une *sorte* $\sigma(f)$ dans $\mathscr{S}$.

On appelle *rang* de f la longueur $\rho(f)$ de $\alpha(f)$, c'est-à-dire le nombre d'arguments de f. Il peut être nul dans le cas d'une *constante* (ou d'une *variable*).

On dit que F est *gradué* si $\mathscr{S}$ est singleton, c'est-à-dire si chaque élément de f est seulement muni d'un rang.

1.2. Magmas hétérogènes

Soit F une $\mathscr{S}$-signature. Un *F-magma hétérogène* est un objet :

$$M = \langle\langle M_s \rangle s \in \mathscr{S}, \langle f_M \rangle f \in F \rangle,$$

où M_s est un ensemble, le domaine de M de sorte s, et f_M est une fonction totale :

$$M_{s_1} \times \ldots \times M_{s_n} \to M_s \quad \text{où } \alpha(f) = s_1 s_2 \ldots s_n \quad \text{et} \quad \sigma(f) = s.$$

Un *homomorphisme* de F-magmas (hétérogènes) h: $M \to M'$ est une application qui préserve les sortes et les opérations d'une manière évidente.

Le *F-magma* initial s'identifie au F-magma des termes bien formés vis-à-vis des sortes et des arités, noté $M(F)$ [on note $M(F)_s$ l'ensemble des termes de sorte s]. Ces termes sont encore identifiés à des arbres finis d'une manière classique.

Exemple : succ désignant la fonction successeur, on a :

$$M(\{0, \text{succ}\}) = \{\text{succ}^n(0)/n \in \mathbf{N}\}.$$

Un terme t de $M(F)_s$ définit dans tout F-magma M un élément de M_s noté $\mathbf{eval}_M(t)$. En d'autres termes, $\mathbf{eval}_M$ est l'unique homomorphisme : $M(F) \to M$. Si $X = (X_s)_{s \in \mathscr{S}}$ est une famille d'ensembles de variables [$x \in X_s$ sera dite de *sorte* s notée $\sigma(x)$], on notera $M(F, X)$ le *F-magma (hétérogène) libre engendré par F*. En fait, $M(F, X) = M(F \cup X)$.

Les éléments de $M(F, X)$ seront représentés par des arbres dont les nœuds seront étiquetés par les symboles f de F ayant un rang $\rho(f) \geqq 1$ et les feuilles par les variables et les symboles de F de rang 0 (constantes).

A tout terme t de $M(F, \{x_1, \ldots, x_k\})$ on associe une fonction $\mathbf{derop}_M(t)$:

$$M_{s_1} \times \ldots \times M_{s_k} \to M_s \quad \text{où } s_i = \sigma(x_i) \quad \text{et} \quad s = \sigma(t),$$

pour tout magma hétérogène M.

1.3. Notations et opérations concernant les arbres et les termes

On note $\mathbf{Var}(t)$ l'ensemble des variables ayant au moins une occurrence dans t.

Un terme t est *linéaire* si chaque variable a au plus une occurrence dans t.

On note $t[u_1/x_1, \ldots, u_k/x_k]$ le résultat de la substitution simultanée du terme u_i à chaque occurence de la variable x_i pour tout $i=1, \ldots, k$, ou encore $t[u_1, \ldots, u_k]$ lorsque la suite $(x_1, \ldots, x_k)$ est connue par le contexte.

Si $G \subseteq F$ et $T \subseteq M(F, X)$ on note $G(T)$ l'ensemble des termes de la forme $g(t_1, \ldots, t_k)$ tels que $g \in G$, $t_1, \ldots, t_k \in T$ et $M(F, T)$ l'ensemble des termes de la forme $u[t_1/x_1, \ldots, t_k/x_k]$ tels que :

$$u \in M(F, \{x_1, \ldots, x_k\}), \quad k \geqq 0 \qquad \text{et} \qquad t_1, \ldots, t_k \in \mathrm{T}.$$

1.4. Définitions récursives par cas

Soit $F=\{f, g, h, \ldots\}$ et $\Phi=\{\varphi, \psi, \ldots\}$ deux alphabets finis gradués; on supposera que $\rho(\varphi) \geqq 1$ pour tout $\varphi \in \Phi$.

Soit $X=\{x_1, x_2, \ldots, x_n, \ldots\}$ l'ensemble des variables que nous utiliserons (et $X_k=\{x_1, \ldots, x_k\} \subset X$).

Un *schéma récursif par cas* est un couple $S=(\Sigma, t)$ formé d'un terme t, élément de $M(F \cup \Phi, X)$ et d'un *système de définitions récursives par cas* Σ, c'est-à-dire d'un ensemble fini de couples (s, u) appelés *clauses*, tels que :

1.4.1. s est linéaire et appartient à $\Phi(M(F, X))$.

1.4.2. u appartient à $M(F \cup \Phi, X)$ et $\mathbf{Var}(u) \subseteq \mathbf{Var}(s)$.

Dans la suite on parlera plus simplement d'un *schéma* $S=(\Sigma, t)$ et de son *système* sous-jacent Σ.

On associe classiquement à Σ une relation semi-thuienne $\xrightarrow[\Sigma]{*}$ sur $M(F \cup \Phi, X)$. *Voir* par exemple Huet [14].

Une *interprétation* est un F-magma I. Il en résulte en particulier que les fonctions f_I sont totales et strictes.

Lorsque l'ensemble des *fonctions de base*, i. e. F, n'est pas connu par le contexte, on le précise en parlant de *F-schéma* et de *F-interprétation.*

Nous appellerons *schéma interprété* (resp. *système interprété*) le couple formé d'un schéma S et d'une interprétation I convenable (resp. d'un système et d'une interprétation).

Les schémas définis ci-dessus sont *non-déterministes* : plus précisément, à tout schéma $S=(\Sigma, t)$, à toute interprétation I de domaine D, à tous $d_1, \ldots, d_k$ (où l'on suppose que $\mathbf{Var}(t) \subseteq \{x_1, \ldots, x_k\}$), on associe un *ensemble de valeurs* possibles noté $\mathbf{Res}_I(S)(d_1, \ldots, d_k)$ (ou encore $\mathbf{Res}_I(t)(d_1, \ldots, d_k)$ lorsque Σ est connu par le contexte).

Nous allons maintenant définir précisément $\mathbf{Res}_I(S)(d_1, \ldots, d_k)$, de façon opérationnelle c'est-à-dire au moyen d'une notion de calcul.

1.5. Calculs

Les calculs vont se faire par réécriture sur des éléments de $M(F \cup \Phi, D)$, c'est-à-dire sur des termes faisant intervenir :

- des *symboles* de procédures (les éléments de Φ);
- des *symboles* de fonction de base (les éléments de F);
- des *éléments* du domaine de calcul D.

Un élément w de $M(F \cup \Phi, D)$ pourra toujours s'écrire $t[d_1, \ldots, d_k]$ pour quelque t dans $M(F \cup \Phi, X_k)$, $d_1, \ldots, d_k \in D$.

On utilisera la relation $\xrightarrow[\Sigma]{*}$ évoquée ci-dessus ainsi que la relation semi-Thuienne engendrée par l'ensemble de paires $R(I)$ suivant :

$$\left.\begin{array}{c}(f(d_1, \ldots, d_n), d), \\ \text{pour tout } f \in F \text{ de rang } n, \text{ tous } d, d_1, \ldots, d_n \text{ dans } D \\ \text{tels que } d = f_I(d_1, \ldots, d_n).\end{array}\right\} \quad (1.5.1)$$

On notera :

$\xrightarrow[I]{}$ la relation de réécriture sur $M(F \cup \Phi, D)$ associée à $R(I)$;

$\xrightarrow[I]{*}$ sa fermeture réflexive et transitive;

$\xleftrightarrow[I]{} = \xrightarrow[I]{} \cup \xleftarrow[I]{}$;

$\xleftrightarrow[I]{*}$ sa fermeture réflexive, symétrique et transitive; c'est la plus petite congruence sur $M(F \cup \Phi, D)$ contenant $R(I)$.

1.5.2. LEMME : *La relation* $\xrightarrow[I]{}$ *est Noetherienne et confluente sur* $M(F \cup \Phi, D)$.

Preuve : Conséquence immédiate des résultats de Huet [14]. □

Il en résulte que tout élément t de $M(F \cup \Phi, D)$ a une unique forme normale pour $\xrightarrow[I]{}$, sa *I-forme normale*, notée $\mathbf{nf}_I(t)$.

Plusieurs types de suites de calculs peuvent être considérés.

Le plus large, formalisant l'appel par nom, utilisera la relation $\xrightarrow[\Sigma, I]{} = \xrightarrow[\Sigma]{} \cup \xleftrightarrow[I]{}$.

L'autre, qui formalise l'*appel par valeur*, utilisera la relation :

$$\underset{\Sigma, I}{\rightarrow}' = \underset{\Sigma}{\rightarrow}' \cup \underset{I}{\leftrightarrow} \qquad \text{où} \quad \underset{\Sigma}{\rightarrow}'$$

ne réécrit φ dans $\varphi(w_1, \ldots, w_k)$ que si $w_1, \ldots, w_k \in M(F, D)$.

Formellement, $\underset{\Sigma}{\rightarrow}'$ est la relation sur $M(F \cup \Phi, D)$ engendrée par l'ensemble $R'(\Sigma)$ des paires de la forme :

$$s[w_1, \ldots, w_k] \rightarrow u[w_1, \ldots, w_k] \qquad \text{pour} \quad (s, u) \in \Sigma,$$
$$\text{avec } \mathbf{Var}(s) \subseteq X_k,\ w_1, \ldots, w_k \in M(F, D).$$

On définira donc :

$$\mathbf{Res}_I^N(t[d_1, \ldots, d_k]) = \{d \in D / t[d_1, \ldots, d_k] \overset{*}{\underset{\Sigma, I}{\rightarrow}} d\},$$

$$\mathbf{Res}_I^V(t[d_1, \ldots, d_k]) = \{d \in D / t[d_1, \ldots, d_k] \overset{*}{\underset{\Sigma, I}{\rightarrow}}' d\}.$$

(On notera aussi $\mathbf{Res}_I^N(t)(d_1, \ldots, d_k)$ au lieu de $\mathbf{Res}_I^N(t[d1, \ldots, d_k])$ et de même pour $\mathbf{Res}_T^V$.)

Remarquer que $\mathbf{Res}_I^V(t)(d_1, \ldots, d_k)$ peut être vide [$t(d_1, \ldots, d_k)$ non défini dans I], singleton [$t(d_1, \ldots, d_k)$ défini de manière unique dans I] ou à plus d'un élément [$t(d_1, \ldots, d_k)$ a plusieurs valeurs dans I résultant d'un non-déterminisme].

On a toujours :

$$\mathbf{Res}_I^V(t[d_1, \ldots, d_k]) \subseteq \mathbf{Res}_I^N(t[d_1, \ldots, d_k])$$

et l'inclusion peut être stricte comme le montre l'exemple suivant :

1.5.3. *Exemple :*

Soit Σ le système ci-dessous et I une interprétation quelconque :

$$\left.\begin{array}{l} \varphi(f(x), y) \rightarrow x, \\ \psi(x) \rightarrow \psi(x). \end{array}\right\}$$

Soit $t = \varphi(f(d_1), \psi(d_2))$; *alors :*

$$\mathbf{Res}_I^V(t) = \varnothing,$$
$$\mathbf{Res}_I^N(t) = \{d_1\}.$$

Dans la suite, on s'intéressera exclusivement à $\mathbf{Res}_I^V(t)$ que l'on notera simplement $\mathbf{Res}_I$.

Voici quelques lemmes techniques relatifs à $\mathbf{Res}_I$ et à $\underset{\Sigma, I}{\overset{*}{\rightarrow}}{}'$.

Le lecteur attentif notera que $\mathbf{Res}_I(t)$ dépend de l'ensemble des variables que l'on considère dans t. On écrira $\mathbf{Res}_{I,k}(t)$ pour préciser que cet ensemble est $\{x_1, \ldots, x_k\}$.

1.5.4. LEMME : *La fonction* $\mathbf{Res}_{I,k}: M(F \cup \Phi, X_k) \to (\mathbf{D}^k \to \mathscr{P}(D))$ *peut se définir par l'induction suivante :*

1.5.4.1. $\mathbf{Res}_{I,k}(x_i)(d_1, \ldots, d_k) = \{d_i\}$.

1.5.4.2. $\mathbf{Res}_{I,k}(f(t_1, \ldots, t_n))\ (d_1, \ldots, d_k)$
$$= \{f_I(d'_1, \ldots, d'_n)/d'_i \in \mathbf{Res}_{I,k}(t_i)(d_1, \ldots, d_k)\}.$$

1.5.4.3. $\mathbf{Res}_{I,k}(\varphi(t_1, \ldots, t_n))\ (d_1, \ldots, d_k)$
$$= \cup \{\mathbf{Res}_{I,n}(\varphi(x_1, \ldots, x_n))(d'_1, \ldots, d'_n)/d'_i \in \mathbf{Res}_{I,k}(t_i)(d_1, \ldots, d_k)\}.$$

Il en résulte que $\mathbf{Res}_{I,k}(f(t_1, \ldots, t_n))\ (d_1, \ldots, d_k)$ est vide si l'un des $\mathbf{Res}_{I,k}(t_i)\ (d_1, \ldots, d_k)$ est vide (pour $f \in F \cup \Phi$).

REMARQUE : L'équation 1.5.4.3 n'est pas vérifiée pour $\mathbf{Res}_I^N$. □

Nous allons donner une autre définition de $\mathbf{Res}_I(t)$, au moyen de réécritures légèrement plus restrictives que celles autorisées par $\underset{\Sigma, I}{\rightarrow}{}'$.

Soit $\underset{\Sigma, I}{\twoheadrightarrow}$ la relation sur $M(F \cup \Phi, D)$ engendrée par l'ensemble $R(\Sigma, I)$ des paires :

$$\left.\begin{array}{c} (\varphi(d_1, \ldots, d_n),\quad t[d'_1/x_1, \ldots, d'_m/x_m]), \\ \text{où } (\varphi(u_1, \ldots, u_n), t) \in \Sigma, \\ u_1, \ldots, u_n \in M(F, X_m),\quad t \in M(F \cup \Phi, X_m), \\ d_i = u_{i_I}(d'_1, \ldots, d'_m) \quad \text{pour } i = 1, \ldots, n, \\ d_1, \ldots, d_n,\quad d'_1, \ldots, d'_m \in D. \end{array}\right\} \qquad 1.5.5$$

On notera $\underset{\Sigma, I}{\Rightarrow}$ la relation $\underset{\Sigma, I}{\twoheadrightarrow} \cup \underset{I}{\rightarrow}$.

1.5.6. LEMME : *Pour tout* t *dans* $M(F \cup \Phi, D)$ *et tout* d *dans* D, $t \underset{\Sigma, I}{\overset{*}{\Rightarrow}} d$ *si et seulement si* $t \underset{\Sigma, I}{\overset{*}{\rightarrow}}{}' d$.

Preuve : 1. Il est clair que :

$$\underset{\Sigma, I}{\twoheadrightarrow} \subseteq \overset{*}{\underset{I}{\leftarrow}} . \underset{\Sigma}{\rightarrow'} \quad \text{et donc} \quad \overset{*}{\underset{\Sigma, I}{\Rightarrow}} \subseteq \left(\overset{*}{\underset{I}{\leftrightarrow}} \cup \underset{\Sigma}{\rightarrow'} \right)^* = \overset{*}{\underset{\Sigma, I}{\rightarrow'}}.$$

2. Réciproquement, soit $t \overset{*}{\underset{\Sigma, I}{\rightarrow'}} d$, autrement dit :

$$t = t_0 \overset{*}{\underset{I}{\leftrightarrow}} t_1 \underset{\Sigma}{\rightarrow'} t_2 \overset{*}{\underset{I}{\leftrightarrow}} t_3 \underset{\Sigma}{\rightarrow'} t_4 \ldots \underset{\Sigma}{\rightarrow'} t_{2n} \overset{*}{\underset{I}{\leftrightarrow}} t_{2n+1} = d.$$

On peut remplacer cette suite de réécritures par :

$$t = t_0 \overset{*}{\underset{I}{\rightarrow}} \mathbf{nf}(u_0) \underset{\Sigma, I}{\twoheadrightarrow} u_2 \overset{*}{\underset{I}{\rightarrow}} \mathbf{nf}(u_2) \underset{\Sigma, I}{\twoheadrightarrow} u_4 \overset{*}{\underset{I}{\rightarrow}} \ldots \underset{\Sigma, I}{\twoheadrightarrow} u_{2n} \overset{*}{\underset{I}{\rightarrow}} d,$$

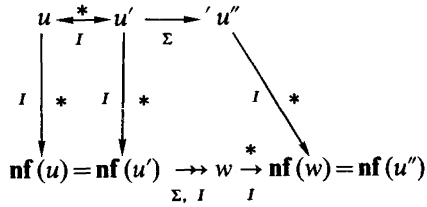
avec :

$$\mathbf{nf}(u_{2i}) = \mathbf{nf}(t_{2i}) \qquad \text{pour} \quad i = 0 \text{ à } n,$$

en vertu du lemme (1.5.7) qui suit (et **nf** désigne $\mathbf{nf}_I$). □

1.5.7. LEMME : *Pour tous u, u', u'' dans* $M(F \cup \Phi, D)$ *tels que* $u \overset{*}{\underset{I}{\leftrightarrow}} u' \underset{\Sigma}{\rightarrow'} u''$, *il existe w tel que :*

$\mathbf{nf}(u) \underset{\Sigma, I}{\twoheadrightarrow} w \overset{*}{\underset{I}{\rightarrow}} \mathbf{nf}(w) = \mathbf{nf}(u'')$. Ce que l'on peut illustrer ainsi :



Preuve : Par induction sur la structure de u'.

Les seuls cas à considérer sont :

1er *cas :*

$u' = \alpha[u'_1, \ldots, u'_k]$, $\quad u'_1, \ldots, u'_k \in M(F, D)$, $\quad u'' = \beta[u'_1, \ldots, u'_k]$, $\quad (\alpha, \beta) \in \Sigma$.

On prend alors :

$$w = \beta\,[\mathbf{nf}\,(u'_1),\ \ldots,\ \mathbf{nf}\,(u'_k)],$$

2e *cas :*

$u' = f(u'_1, \ldots, u'_k)$, $f \in F \cup \Phi, u'_1, \ldots, u'_k \in M(F \cup \Phi, D)$, $u'' = f(u''_1, \ldots, u''_k)$

et il existe i tel que :

$$u'_i \underset{\Sigma}{\rightarrow'} u''_i \qquad \text{et} \qquad u''_j = u'_j \quad \text{pour } j \neq i.$$

On prend alors $w = f(w_1, \ldots, w_k)$ avec $w_j = \mathbf{nf}(u''_j)$ pour $j \neq i$ et w_i associé à (u'_i, u''_i) par hypothèse d'induction. □

1.6. Déterminisme, confluence et cohérence

Un système interprété (Σ, I) est *déterministe* si pour tout $\varphi \in \Phi$ et tout $(d_1, \ldots, d_n)$ dans D^n [où $n = \rho(\varphi)$], il existe au plus une règle $(\varphi(u_1, \ldots, u_n), t)$ dans Σ et au plus un m-uplet $(d'_1, \ldots, d'_m)$ qui satisfont (1.5.5).

Remarquer que cette condition ne dépend pas des membres droits des règles de Σ. Ce n'est pas le cas des conditions que nous définissons maintenant.

Un système interprété (Σ, I) est *confluent* si la relation $\underset{\Sigma, I}{\Rightarrow}$ est confluente (*cf.* Huet [14]), c'est-à-dire si, pour tous t_1, t_2, t_3 tels que :

$$t_1 \overset{*}{\underset{\Sigma, I}{\Rightarrow}} t_2 \qquad \text{et} \qquad t_1 \overset{*}{\underset{\Sigma, I}{\Rightarrow}} t_3,$$

il existe t_4 tel que :

$$t_2 \overset{*}{\underset{\Sigma, I}{\Rightarrow}} t_4 \qquad \text{et} \qquad t_3 \overset{*}{\underset{\Sigma, I}{\Rightarrow}} t_4.$$

La preuve de (1.5.6) permet de montrer que $\underset{\Sigma, I}{\Rightarrow}$ est confluente si et seulement si $\underset{\Sigma, I}{\rightarrow'}$ est confluente.

Un système interprété (Σ, I) est *cohérent* si pour tout φ dans Φ et tout $(d_1, \ldots, d_n)$ dans D^n, l'ensemble $\mathbf{Res}_I(\varphi(d_1, \ldots, d_n))$ a au plus un élément.

Il résulte alors du lemme (1.5.4) que $\mathbf{Res}_I(t)$ a au plus un élément pour tout t dans $M(F \cup \Phi, D)$, autrement dit que $\mathbf{Res}_{I,k}$ est une fonction : $M(F \cup \Phi, X_k) \rightarrow (D^k \rightarrow D)$.

1.6.1. PROPOSITION : *Pour tout système interprété* (Σ, I) :

(i) *confluent implique cohérent;*

(ii) *déterministe implique confluent.*

Les implications inverses ne sont pas vraies.

Preuve :

(i) Montrons que : non-cohérent implique non-confluent.

Si (Σ, I) est non-cohérent, alors :

Il existe φ et $(d_1, \ldots, d_n)$ tels que :

$$\varphi(d_1, \ldots, d_n) \underset{\Sigma, I}{\overset{*}{\Rightarrow}} e_1 \qquad \text{et} \qquad \varphi(d_1, \ldots, d_n) \underset{\Sigma, I}{\overset{*}{\Rightarrow}} e_2.$$

Puisque e_1 et e_2 sont deux éléments différents de D, il n'existe pas de terme t_3 tel que :

$$e_1 \underset{\Sigma, I}{\overset{*}{\Rightarrow}} t_3 \qquad \text{et} \qquad e_2 \underset{\Sigma, I}{\overset{*}{\Rightarrow}} t_3.$$

Donc (Σ, I) n'est pas confluent.

Implication inverse : contre-exemple

Considérons le système suivant, interprété sur les entiers naturels :

$$\psi(x) = 1 + \psi(x),$$
$$\psi(x \,.\, y) = 1 + \psi(x),$$
$$\psi(x \,.\, y) = 1 + \psi(y).$$

Ce système interprété est cohérent de manière triviale car aucun $\psi(d)$ n'a pas de valeur.

Il est d'autre part non-confluent, car le diagramme suivant n'obéit pas à la définition de la confluence :

$$\begin{array}{l} \psi(2.3) \to 1 + \psi(2), \\ \qquad\quad \searrow\ 1 + \psi(3). \end{array}$$

(ii) Montrons que : déterministe implique confluent.

Si (Σ, I) est déterministe, la relation de réécriture $\underset{\Sigma, I}{\Rightarrow} = \underset{\Sigma, I}{\to\!\!\!\to} \cup \underset{I}{\to}$ est sans paires critiques (*cf.* Huet [14]) et ce de manière triviale puisqu'il n'y a pas de variables.

La confluence résulte de Rosen [24].

L'implication inverse n'est pas vraie; il suffit de considérer le système :

$$\begin{cases} \psi(x_1) \to x_1, \\ \psi(f(x_1)) \to f(x_1). \end{cases}$$

qui est confluent mais non déterministe. □

1.7. Terminaison

Un système interprété (Σ, I) est *total* si tout terme t dans $M(F \cup \Phi, D)$ a au moins une valeur i. e. si $\mathbf{Res}_I(t)(d_1, \ldots, d_k) \neq \varnothing$. [Il suffit pour assurer cela que tout terme de la forme $\varphi(d_1, \ldots, d_n)$ ait au moins une valeur, encore d'après le lemme (1.5.4).]

La totalité pourra être assurée au moyen des propriétés suivantes.

Un système (Σ, I) est *Noetherien* si la relation $\underset{\Sigma, I}{\Rightarrow}$ est *Noetherienne* i. e. s'il n'existe pas de suite infinie $(t_i)i \in \mathbb{N}$ où $t_i \in M(F \cup \Phi, D)$ telle que $t_i \underset{\Sigma, I}{\Rightarrow} t_{i+1}$ pour tout $i \geqq 0$.

Un système (Σ, I) est *partout défini* si pour tout ψ dans Φ et tout $(d_1, \ldots, d_n)$ dans D^n [avec $n = \rho(\psi)$], $\psi(d_1, \ldots, d_n) \underset{\Sigma, I}{\to\!\!\!\!\to} t$ pour quelque $t \in M(F \cup \Phi, D)$.

Remarquer que cette dernière propriété ne dépend que des membres gauches des règles de Σ.

Les relations entre ces diverses propriétés sont réunies dans la proposition suivante :

1.7.1. Proposition : *Pour un système interprété* (Σ, I) *:*

1. *total implique partout défini; la réciproque n'est pas vraie;*
2. *partout défini et Noethérien implique total; la réciproque n'est pas vraie.*

Ces résultats sont classiques et valent dans des cas plus généraux (Huet [14]).

Preuve :

1. Évident.
2. Soit t dans $M(F \cup \Phi, D)$. Il existe nécessairement une suite de réécritures $t \underset{\Sigma, I}{\overset{*}{\Rightarrow}} u$ telle que u soit irréductible puisque le schéma est Noetherien. On veut montrer que $u \in D$. Si $u \notin D$, u est un terme; il contient au moins un sous-terme de la forme $f(d_1, \ldots, d_k)$ qui se réduit en $d = f_I(d_1, \ldots, d_k)$ car

f_I est totale, ou de la forme $\psi(d_1, \ldots, d_k)$ qui se réduit car (Σ, I) est supposé partout défini. Donc u est irréductible.

Donnons un exemple d'un système interprété qui est total et partout défini (et même confluent) mais non Noetherien :

$$\left.\begin{array}{l} \psi(x_1) \to \psi(x_1), \\ \psi(x_1) \to a. \end{array}\right\}$$

Il admet un calcul infini :

$$\psi(d) \to \psi(d) \to \psi(d) \ldots \to \psi(d) \to \ldots$$

1.7.2. PROPOSITION : *Un système total est cohérent si et seulement si il est confluent.*

Soit (Σ, I) un système interprété.

1. Montrons que, si (Σ, I) est total et cohérent, alors il est confluent.

Soit t dans $M(F \cup \Phi, D)$.

Supposons :

$$t \underset{\Sigma, I}{\overset{*}{\Rightarrow}} t_1 \qquad \text{et} \qquad t \underset{\Sigma, I}{\overset{*}{\Rightarrow}} t_2.$$

(Σ, I) étant total, il existe $e_1 \in D$ et $e_2 \in D$ tels que :

$$t_1 \underset{\Sigma, I}{\overset{*}{\Rightarrow}} e_1 \qquad \text{et} \qquad t_2 \underset{\Sigma, I}{\overset{*}{\Rightarrow}} e_2.$$

e_1 et e_2 sont donc deux valeurs de t; elles doivent être égales puisque (Σ, I) est supposé cohérent.

D'où la confluence.

2. Réciproque : c'est une conséquence de la proposition (1.6.1) [point (i)]. □

1.8. Schémas récursifs par cas avec sortes

Tout ce qui a été défini ci-dessus (§ 1.4-1.7) s'étend de manière triviale au cas où l'on utilise des signatures F et Φ et non plus seulement des alphabets gradués. On est alors conduit à utiliser également des ensembles de variables avec sortes. Les deux membres d'une même clause doivent être de même sorte, etc.

Les F-magnas hétérogènes forment alors la classe d'interprétations convenable.

Mentionnons que l'exemple (0.1) utilisait deux sortes : la sorte « mot » et la sorte « état ».

2. EXEMPLES

On donne une série d'exemples destinés à illustrer les définitions précédentes. Les preuves des affirmations données ci-dessous pourront être faites au moyen des résultats des sections suivantes.

2.1. Fonctions primitives récursives

On notera de la même façon l'ensemble **N** des entiers $\geqq 0$ et le F-magma $\langle \mathbf{N}, 0, 1, +, * \rangle$ où $F=\{0, 1, +, *\}$ (on ne fait donc pas de distinction entre la fonction ou la constante et le symbole qui la représente).

On écrira donc la fonction factorielle au moyen du système suivant :

$$\left.\begin{aligned} \psi(0) &= 1, \\ \psi(x+1) &= (x+1) * \psi(x). \end{aligned}\right\}$$

Il est clair que $\psi_{\mathbf{N}}(x)=x!$ pour tout $x \in \mathbf{N}$.

Plus généralement, toute fonction primitive récursive peut être définie au moyen d'un F-schéma interprété déterministe et total (dont **N** est l'interprétation).

2.2. Fonction d'Ackermann

Voir l'exemple (0.2).

2.3. Fonctions sur $\mathbf{N} \times \mathbf{N}$

Les exemples précédents reposent sur l'isomorphisme : $\langle \mathbf{N}, 0, s \rangle \to M(\{\mathbf{0}, s\})$ où s désigne la fonction successeur : $\lambda x \in \mathbf{N} . x+1$.

Considérons maintenant $\langle \mathbf{N} \times \mathbf{N}, (0, 0), s_1, s_2 \rangle$ où :

$$s_1 = \lambda (x, y) \in \mathbf{N} \times \mathbf{N} . (x+1, y) \qquad \text{et} \qquad s_2 = \lambda (x, y) \in \mathbf{N} \times \mathbf{N} . (x, y+1).$$

Nous utiliserons le symbole **0** pour désigner l'élément (0, 0) de $\mathbf{N} \times \mathbf{N}$.

Il est à remarquer que $s_1(s_2(z)) = s_2(s_1(z))$ pour tout z dans $\mathbf{N} \times \mathbf{N}$ et, plus précisément, que $\langle \mathbf{N} \times \mathbf{N}, (0, 0), s_1, s_2 \rangle$ est isomorphe au magma quotient $M(\{s_1, s_2, \mathbf{0}\})/R$ où R est la congruence engendrée par la relation $s_1(s_2(z)) = s_2(s_1(z))$.

La fonction partielle $\lambda(x, y) \in \mathbf{N} \times \mathbf{N}. (x+y, x-y)$ avec $x \geqq y$ peut se définir ainsi par le système *cohérent* $(\Sigma, \mathbf{N} \times \mathbf{N})$:

$$\theta(\mathbf{0}) = \mathbf{0},$$
$$\theta(s_1\, x) = s_1\, s_2\, \theta(x),$$
$$\theta(s_1\, s_2\, x) = s_1\, s_1\, \theta(x).$$

2.4. Fonction de transition d'un automate fini

Voir l'exemple (0.1). Remarquer l'utilisation (implicite) de deux sortes « mot » et « état ».

2.5. Comparaison de deux termes

Soit G_0 un alphabet fini gradué, V un ensemble fini de variables, Ω le symbole de constante servant à définir la relation d'approximation $\leqq$ sur $M(G)$, où $G = G_0 \cup V \cup \{\Omega\}$.

(*Cf.* Courcelle [3] ou [4]).

Pour tous symboles f, f' dans G tels que $f \neq f'$, introduisons :

un symbole a_f d'arité $\rho(f)$;

un symbole $b_{f,f'}$ d'arité $\rho(f) + \rho(f')$.

Soit $H = \{a_m, b_{m,m'} / m, m' \in G, m \neq m'\}$.

Considérons maintenant le système :

$$\left.\begin{aligned} \psi(f(x_1, \ldots, x_n), f(y_1, \ldots, y_n)) &= a_f(\psi(x_1, y_1), \ldots, \psi(x_n, y_n)), \\ \psi(f(x_1, \ldots, x_n), f'(y_1, \ldots, y_m)) &= b_{f,f'}(x_1, \ldots, x_n, y_1, \ldots, y_m), \end{aligned}\right\}$$

pour tous $f, f' \in G$ tels que $f' \neq f$, avec $n = \rho(f)$ et $m = \rho(f')$.

Nous allons en donner plusieurs interprétations différentes.

1re *interprétation : Vérifier l'égalité de deux termes*

Il existe une interprétation I_1 telle que :

$\psi_{I_1}(t, t') = si\ t = t'\ alors\ vrai\ sinon\ faux$ pour tous $t, t' \in M(G)$.

Il suffit de prendre, pour $b_1, \ldots, b_n \in \{vrai, faux\}$:

(1) $a_f(b_1, \ldots, b_n) = b_1\ et\ b_2\ et\ \ldots\ et\ b_n$ si $\rho(f) = n \geqq 1$,

$a_f = vrai$ si $\rho(f) = 0$;

(2) $b_{f,g}(t_1, \ldots, t_{n+m}) = faux$.

2e *interprétation : Vérifier que* $t \leqq t'$

Il existe I_2 telle que :

$\Psi_{I_2}(t, t') = si\ t \leqq t'\ alors\ vrai\ sinon\ faux.$

Il suffit de prendre les $a_f \cdot s$ comme ci-dessus et :

(2') $b_{\Omega, g}(t_1, \ldots, t_m) = vrai,$

(2'') $b_{f, g}(t_1, \ldots, t_{n+m}) = faux,$

si $f \neq \Omega$ [*et* $\rho(f) = n$].

2.6. Schémas primitifs récursifs avec paramètres

Ces schémas ont été introduits dans Courcelle et Franchi-Zannettacci ([6], [29]) pour calculer les valeurs des attributs synthétisés définis à la racine d'un arbre de dérivation de la grammaire sous-jacente, en fonction des valeurs des attributs hérités (également Mayoh [21]; Engelfriet [28]). Ce sont des exemples de schémas totaux et déterministes.

2.7. Sémantique dénotationnelle

La méthode de définition de la sémantique des langages de programmation connue sous le nom de *sémantique dénotationnelle* ([25], [26]) utilise des définitions récursives par cas, dont les arguments sont des mots dans la présentation traditionnelle, mais en fait des arbres de dérivation ainsi qu'il est montré dans [12], p. 76-79.

2.8. Types de données abstraits algébriques

Certains calculs sur des quotients du magma initial sont formalisables au moyen de définitions récursives par cas. Voir à ce sujet les exemples donnés dans Huet et Hullot [15].

2.9. PROLOG

Enfin, si la parenté avec le langage PROLOG est indéniable, il ne faut pas manquer de mettre en évidence les différences :

– notre formalisme est fonctionnel alors que celui de PROLOG est relationnel (les fonctions définies par programmes le sont par relations);

– l'ordre des clauses importe en PROLOG mais non ici;

– nous n'introduisons rien qui ressemble à l'opérateur de coupure (« cut » ou « slash »).

On peut donc affirmer que nos schémas récursifs par cas correspondent à certains programmes PROLOG assez particuliers, et que l'étude que nous

en faisons constitue une étape dans la formalisation de la sémantique de PROLOG, ou d'un dérivé fonctionnel.

3. PLUS PETITS POINTS FIXES

Jusqu'à présent, nous avons défini uniquement de manière opérationnelle la sémantique des schémas récursifs par cas.

La raison en est que cette méthode est bien adaptée aux schémas considérés en ce sens qu'un système *est* déjà dans sa syntaxe un système de réécriture.

Nous nous proposons maintenant de l'étudier en termes de plus petits points fixes, ne serait-ce que pour faire le lien avec les théories établies.

Nous nous limiterons au cas des systèmes cohérents, laissant pour une étude ultérieure le cas des systèmes *vraiment* non déterministes, qui présentent des difficultés particulières déjà rencontrées par Boudol [2] et d'autres sous divers aspects.

Soit (Σ, I) un système interprété cohérent. Pour tout φ dans Φ d'arité n, on désignera par φ_I la fonction partielle $\mathbf{Res}_I(\varphi(x_1, \ldots, x_n)) : D^n \to D$. Il en est de même de t_I (D est le domaine de I) si t appartient à $M(F \cup \Phi, X_n)$.

3.1. Définition de l'opérateur dérivé d'un terme

Soit un Φ-uplet de fonctions $\bar{\Phi} = (\bar{\varphi})_{\varphi \in \Phi}$, où $\bar{\varphi}$ est une fonction partielle : $D^{\rho(\varphi)} \to D$ pour tout $\varphi \in \Phi$.

On note $\mathbf{derop}_{\bar{\Phi}, I}$ la fonction qui associe à t dans $M(F \cup \Phi, X_n)$ la fonction partielle : $D^n \to D$ définie par t en prenant f_I comme valeur de f pour tout f dans F et $\bar{\varphi}$ comme valeur de φ pour tout φ dans Φ.

3.2. LEMME : $t_I = \mathbf{derop}_{\bar{\Phi}, I}(t)$.

Ce lemme n'est qu'une reformulation du lemme (1.5.4).

On va considérer Σ comme un système d'équations fonctionnelles et $\Phi_I = (\varphi_I)_{\varphi \in \Phi}$ comme une solution éventuelle de Σ.

3.3. Notions de solution, de sur-solution

Un Φ-uplet $\bar{\Phi} = (\bar{\varphi})_{\varphi \in \Phi}$ de fonctions (de type convenable) est :

une *solution* si $\mathbf{derop}(s) = \mathbf{derop}(t)$ pour toute clause (s, t) dans Σ,

une *sur-solution* si $\mathbf{derop}(s) \supseteq \mathbf{derop}(t)$ pour toute clause (s, t) dans Σ,

où l'on note **derop** la fonction $\mathbf{derop}_{\bar{\Phi}, I}$, et où $\subseteq$ désigne l'ordre sur les fonctions partielles.

Le terme sur-solution est adapté de Manna et Shamir [20].

3.4. THÉORÈME : *Soit* (Σ, I) *un système interprété cohérent.*

1. Φ_I *est la plus petite sur-solution de* (Σ, I).
2. *Si* (Σ, I) *est confluent alors* Φ_I *est sa plus petite solution.*
3. *Si* (Σ, I) *est total (et donc confluent) alors* Φ_I *est l'unique solution de* (Σ, I).

Preuve :

1. Soit $\Phi_I = (\varphi_I)_{\varphi \in \Phi}$ et $\mathbf{derop} = \mathbf{derop}_{\Phi_I, I}$.

Il faut montrer que, pour tout $(s, t) \in \Sigma$, on a $\mathbf{derop}(s) \supseteq \mathbf{derop}(t)$.

Soit $d_1, \ldots, d_k \in D$ tels que $\mathbf{derop}(t)(d_1, \ldots, d_k) = d$.

On a donc $t[d_1, \ldots, d_k] \underset{\Sigma, I}{\overset{*}{\Rightarrow}} d$.

Puisque $(s, t) \in \Sigma$, $s[d_1, \ldots, d_k] \underset{\Sigma}{\rightarrow}' t[d_1, \ldots, d_k]$ et donc $s[d_1, \ldots, d_k] \underset{\Sigma, I}{\overset{*}{\Rightarrow}} d$.

Donc $d = \mathbf{derop}(s)(d_1, \ldots, d_k)$.

Donc Φ_I est une sur-solution de Σ.

Montrons que c'est la plus petite.

Soit $\bar{\Phi}$ une sur-solution arbitraire. Montrons que pour tout u dans $M(F \cup \Phi, X_k)$ on a :

$$\mathbf{derop}(u) \subseteqq \mathbf{derop}_{\bar{\Phi}, I}(u).$$

On fait la preuve par récurrence sur la longueur p d'une suite de réécritures $u[d_1, \ldots, d_k] \underset{\Sigma, I}{\overset{p}{\Rightarrow}} d$ [il faut montrer que $d \in \mathbf{derop}_{\bar{\Phi}, I}(u)(d_1, \ldots, d_k)$]; et pour chaque p, on fait la preuve par récurrence sur la taille de u :

(i) $u = x_i$: $d = d_i = \mathbf{derop}_{\bar{\Phi}, I}(u)$ (indépendamment de $\bar{\Phi}$).

(ii) $u = f \in F_0$: semblable.

(iii) $u = f(u_1, \ldots, u_n)$, $n \geqq 1$ alors $u_i[d_1, \ldots, d_k] \underset{\Sigma, I}{\overset{p_i}{\Rightarrow}} d_i'$ avec $p_i < p$, et la dernière étape de la suite de longueur p est nécessairement $f(d_1', \ldots, d_n') \underset{I}{\rightarrow} d$.

Par hypothèse de récurrence, $d_i' \in \mathbf{derop}_{\bar{\Phi}, I}(u_i)(d_1, \ldots, d_k)$ et donc $d \in \mathbf{derop}_{\bar{\Phi}, I}(f(u_1, \ldots, u_n))$.

(iv) $u = \varphi(u_1, \ldots, u_n)$: on a encore nécessairement :

$$u_i[d_1, \ldots, d_k] \underset{\Sigma, I}{\overset{p_i}{\Rightarrow}} d_i' \qquad \text{pour} \quad i = 1, \ldots, n$$

et la suite de réécritures considérée se termine par :

$$\varphi\,[d'_1, \ldots, d'_n] \underset{\Sigma, I}{\overset{p'}{\Rightarrow}} d \qquad \text{avec} \quad p' \leqq p,$$

(on peut avoir $p_i = 0$ pour tout $i = 1, \ldots, n$).

Cette dernière suite est de la forme :

$$\varphi\,[d'_1, \ldots, d'_n] \underset{\Sigma, I}{\twoheadrightarrow} w\,[d''_1, \ldots, d''_1] \underset{\Sigma, I}{\overset{p''}{\Rightarrow}} d$$

$$w \in M(F \cup \Phi, X_1) \qquad \text{et} \qquad p'' = p' - 1.$$

Par hypothèse de récurrence :

$$d \in \mathbf{derop}_{\bar{\Phi}, I}(w)(d''_1, \ldots, d''_1),$$

d'où l'on conclut, puisque $\bar{\Phi}$ est une sur-solution :

$$d \in \mathbf{derop}_{\bar{\Phi}, I}(\varphi(x_1, \ldots, x_n))(d'_1, \ldots, d'_n).$$

Toujours par récurrence :

$$d'_i \in \mathbf{derop}_{\bar{\Phi}, I}(u_i)(d_1, \ldots, d_k) \quad \text{pour tout } i = 1, \ldots, n$$

donc :

$$d \in \mathbf{derop}_{\bar{\Phi}, I}(\varphi(u_1, \ldots, u_n))(d_1, \ldots, d_k).$$

C.Q.F.D.

REMARQUE : Φ_I n'est pas nécessairement une *solution* de Σ comme le montre l'exemple suivant où $\varphi : \mathbf{N} \times \mathbf{N} \to \mathbf{N}$:

$$\left.\begin{aligned} &\varphi(x, y) = \varphi(x, 0), \\ &\varphi(x+1, y+1) = 1. \end{aligned}\right\} \Sigma.$$

φ_I est la fonction telle que :

$$\varphi_I(x, y) = 1 \qquad \text{si} \quad x > 0, \qquad y > 0, \quad \text{non définie autrement.}$$

Par substitution de φ_I dans la première règle de Σ, on obtient φ_I pour le membre gauche et la fonction partout non définie pour le membre droit. Ainsi φ_I *n'est pas* une solution de Σ.

2. Si (Σ, I) est confluent alors Φ_I est une solution.

Montrons ce point avec les mêmes notations que dans la partie 1.

Soit $(s, t) \in \Sigma$. Soit $s[d_1, \ldots, d_k] \underset{\Sigma, I}{\overset{*}{\Rightarrow}} d$ une suite de réécritures.

On a aussi $s[d_1, \ldots, d_k] \underset{\Sigma}{\rightarrow}' t[d_1, \ldots, d_k]$.

Par la confluence, il existe u dans $M(F \cup \Phi, D)$ tel que :

$$d \underset{\Sigma, I}{\overset{*}{\Rightarrow}} u \qquad \text{et} \qquad t[d, \ldots, d_k] \underset{\Sigma, I}{\overset{*}{\Rightarrow}} u.$$

Puisque $d \in D$, on a nécessairement $u = d$ et donc $d \in \mathbf{derop}(t)(d_1, \ldots, d_k)$. Puisque $\mathbf{derop}(s) \supseteq \mathbf{derop}(t)$ par la première partie, on a égalité.

Soit $\bar{\Phi} = (\bar{\varphi})_{\varphi \in \Phi}$ une autre solution.

Toute solution de (Σ, I) est aussi sur-solution de (Σ, I).

Or, on a montré que Φ_I est la plus petite sur-solution. Donc Φ_I est plus petite que $\bar{\Phi}$.

3. *Si* (Σ, I) *est total (donc confluent), alors* Φ_I *est son unique solution :*

Soit $\bar{\Phi} = (\bar{\varphi})_{\varphi \in \Phi}$ une autre solution, $\varphi_I \in \Phi_I$ et $\bar{\varphi} \in \bar{\Phi}$.

On a donc [par (3.4.2)] $\varphi_I \subseteq \bar{\varphi}$. Puisque φ_I est une fonction totale, $\varphi_I = \bar{\varphi}$. □

3.5. *Exemple :* Donnons un exemple d'un système (Σ, I) confluent ayant plusieurs solutions. On prend encore $I = \mathbf{N}$:

$$\left.\begin{aligned} \varphi(0) &= \varphi(0), \\ \varphi(x+1) &= 2 \,.\, \varphi(x). \end{aligned}\right\}$$

Ce système est confluent, φ_I est partout non définie mais tout choix d'une valeur a pour $\varphi(0)$ définit une unique solution φ_a de Σ telle que $\varphi_a(0) = a$. □

L'intérêt de *l'unicité* de la solution d'un schéma récursif polyadique a été expliqué dans Courcelle [3] et concerne en particulier la preuve de l'équivalence de deux définitions récursives. La méthode fondée sur l'unicité de la solution s'applique également aux schémas récursifs par cas.

4. SYSTÈMES ET SCHÉMAS SEMI-INTERPRÉTÉS

Les systèmes et schémas récursifs par cas étudiés jusqu'ici sont très généraux, et présentent les difficultés inhérentes aux systèmes de réécriture.

Afin de poursuivre une étude théorique la plus générale possible de situations fréquemment rencontrées, nous allons introduire une famille particulière

de systèmes que nous appellerons *spéciaux* et qui comprend l'essentiel des exemples que nous avons donnés dans la section 2.

Les conditions syntaxiques qui caractérisent les systèmes spéciaux s'introduisent commodément au moyen de *sortes* sur les alphabets F et Φ. Elles permettront d'écrire une interprétation I sous forme $J+K$ où J permet « d'interpréter les membres gauches ». En fixant J et en faisant varier K on obtiendra une famille de systèmes interprétés qui ont en commun le couple (Σ, J) (on appellera ce couple *système semi-interprété*).

4.1. Définitions

Nous allons définir les *systèmes spéciaux* au moyen des objets suivants :

un ensemble de *sortes* $\mathscr{S} = \mathscr{T} \cup \mathscr{U}$ où $\mathscr{T}$ et $\mathscr{U}$ sont disjoints;

une $\mathscr{S}$-signature $F = G \cup H$ où $G \cap H = \varnothing$; G est une $\mathscr{T}$-signature et tout élément f de H a une sorte $\sigma(f)$ dans $\mathscr{U}$ et une arité $\alpha(f)$ dans $\mathscr{S}^*$;

une $\mathscr{S}$-signature Φ dont tout élément φ a une sorte $\sigma(\varphi)$ dans $\mathscr{U}$ et une arité $\alpha(\varphi)$ dans $\mathscr{S}^+$ (i.e. $\alpha(\varphi) \in \mathscr{S}^*$ et $\rho(\varphi) \geqq 1$). Par convention on écrira toujours les arguments de sorte $\mathscr{T}$ *avant* les arguments de sorte $\mathscr{U}$, autrement dit on supposera toujours que $\alpha(\varphi) \in \mathscr{T}^* \mathscr{U}^*$.

On utilisera un ensemble $X = (X_s)_{s \in \mathscr{T}}$ et un ensemble $Y = (Y_s)_{s \in \mathscr{U}}$ de variables.

Ces variables joueront des rôles différents :

— les variables de Y, donc de sorte dans $\mathscr{U}$, seront de simples paramètres, dont les *valeurs n'influeront pas sur le déroulement du calcul;*

— les variables de X seront les plus importantes : elles contrôleront le déroulement du calcul; de leurs valeurs dépendra la terminaison. On notera $V = X \cup Y$. Le symbole «;» séparera les deux types d'arguments des fonctions φ.

Un *système spécial* Σ est alors un ensemble de clauses (s, t) dans $M(F \cup \Phi, V)^2$ telles que :

4.1.1. s est linéaire et appartient à $\Phi(M(G, V))$;

4.1.2. t appartient à $M(H \cup \Phi, V)$ et $\mathbf{Var}(t) \subseteqq \mathbf{Var}(s)$;

4.1.3. $\sigma(s) = \sigma(t)$ [et appartient à $\mathscr{U}$ par suite de (4.1.1)].

Les conditions relatives aux sortes font qu'une clause (s, t) est de la forme :

4.1.4. $(\varphi(s_1, \ldots, s_n; y_1, \ldots, y_m), t)$

avec :

(1) $\qquad y_1, \ldots, y_m \in Y \qquad$ et $\qquad s_1, \ldots, s_n \in M(G, X_k)$,

(2) pour tout sous-terme de t de la forme :

$$\psi(t_1, \ldots, t_{n'}; u_1, \ldots, u_{m'}) \qquad (\text{avec } \psi \in \Phi),$$

on a : $t_i \in X_k$ pour tout $i = 1, \ldots, n'$.

Un *schéma spécial* est un couple $S = (\Sigma, \tau)$ formé d'un système spécial et d'un terme $\tau \in M(F \cup \Phi, V)$ de sorte dans $\mathscr{U}$.

Une *interprétation* est alors un F-magma (hétérogène) I.

Une *semi-interprétation* est un G-magma J.

En se restreignant aux domaines $(D_s)_{s \in \mathscr{S}}$ et aux opérateurs $(g_I)_{g \in G}$ on obtient une *semi-interprétation J associée à I.*

On écrira :

$$I = J + K \qquad \text{avec} \quad K = \langle (D_s)_{s \in \mathscr{U}}, (f_I)_{f \in H} \rangle$$

(ce n'est pas en général un H-magma !).

Un *schéma* (resp. un *système*) *semi-interprété* est un couple (S, J) [resp. (Σ, J)] formé d'un schéma spécial (resp. d'un système spécial) et d'une semi-interprétation de type convenable.

Noter que les conditions (4.1.1) à (4.1.3) sont très simples et autorisent l'imbrication des symboles de Φ dans les membres droits des clauses comme dans l'exemple (0.1) (on entend par *imbrication* le fait qu'un symbole de Φ figure dans un terme argument d'un autre symbole de Φ).

4.2. Exemples

Toutes les fonctions primitives récursives [*cf.* (2.1)].

Les exemples (2.3), (2.4), (2.5).

Les schémas primitifs récursifs avec paramètres de [6], [29].

Les définitions de la sémantique dénotationnelle [12, 25, 26, 19].

Dans l'exemple (0.1) (spécial), la variable (de type « mot ») était un argument et la variable q (de type « état ») était un paramètre.

Par contre, dans l'exemple (0.2), aucune des variables x et y ne pouvait être un paramètre et l'imbrication dans le second membre de la troisième clause empêchait le schéma d'être spécial.

4.3. Sémantique; trace d'un calcul dans l'interprétation libre

Soit (Σ, I) un système interprété et $I = J + K$.

On notera $\underset{K}{\rightarrow}$ la relation sur $M(F \cup \Phi, D)$ associée à $(f_I)_{f \in H}$ comme $\underset{I}{\rightarrow}$ était associée à $(f_I)_{f \in F}$.

Au lieu de la relation $\underset{\Sigma, I}{\twoheadrightarrow}$, on utilisera une variante définie comme suit et notée $\underset{\Sigma, J}{\twoheadrightarrow}$:

C'est la relation sur $M(F \cup \Phi, D)$ engendrée par l'ensemble $R'(\Sigma, I)$ des paires de termes suivantes :

$$\left.\begin{array}{c} (\varphi(d_1, \ldots, d_n; u_1, \ldots, u_m), \\ t[d'_1/x_1, \ldots, d'_k/x_k, u_1/y_1, \ldots, u_m/y_m]) \end{array}\right\}$$

telles que :

$$\left.\begin{array}{c} d_1, \ldots, d_n \text{ sont des éléments de } D \text{ de sorte dans } \mathscr{T}, \\ u_1, \ldots, u_m \text{ sont des éléments de } M(H, D) \text{ de sorte dans } \mathscr{U}, \\ d_i = s_{i_J}(d'_1, \ldots, d'_k) \quad \text{pour } i = 1, \ldots, n, \\ (\varphi(s_1, \ldots, s_n; y_1, \ldots, y_m), t) \text{ est une clause de la forme (4.1.4).} \end{array}\right\} \quad (4.3.1)$$

On notera encore $\underset{\Sigma, I}{\Rightarrow}$ la relation $\underset{\Sigma, J}{\twoheadrightarrow} \cup \underset{K}{\rightarrow}$.

Par une preuve très semblable à celle du lemme (1.5.6), on montrerait :

(4.3.2) LEMME : *Pour tout* $u \in M(H \cup \Phi, D)$,

$$Res_{\Sigma, I}(u) = \left\{ d \in D / u \underset{\Sigma, I}{\overset{*}{\Rightarrow}} d \right\}.$$

Nous allons montrer maintenant que toute suite de calculs $u \underset{\Sigma, I}{\overset{*}{\Rightarrow}} d$ peut se mettre sous la forme $u \underset{\Sigma, J}{\overset{*}{\twoheadrightarrow}} u' \underset{K}{\overset{*}{\rightarrow}} d$, ce qui veut dire que l'on peut repousser en une étape indépendante les évaluations utilisant les $(f_I)_{f \in H}$; le terme u' appartient à $M(H \cup \Phi, D)$ et, à un détail de notation près, on appellera $u \underset{\Sigma, J}{\overset{*}{\twoheadrightarrow}} u'$ la *trace* du calcul $u \underset{\Sigma, I}{\overset{*}{\Rightarrow}} d$.

Afin de définir formellement la trace, nous posons les définitions suivantes :

Si I est une interprétation $\langle (D_s)_{s \in \mathscr{S}}, (f_l)_{f \in F} \rangle$ de la forme $J+K$, on désignera par :

D la famille $(D_s)_{s \in \mathscr{S}}$;

D_J la famille $(D_s)_{s \in \mathscr{J}}$;

D_K la famille $(D_s)_{s \in \mathscr{U}}$.

Pour tout élément u de $M(F \cup \Phi, D)$ on peut trouver w dans :

$$M(H \cup \Phi, D_J \cup \{y_1, \ldots, y_m\}) \qquad \text{et} \qquad d_1, \ldots, d_m \in D_K$$

tels que :

$$u = w[d_1/y_1, \ldots, d_m/y_m] \quad \text{(encore noté } w[d_1, \ldots, d_m]).$$

Soit alors un calcul de la forme :

$$u_0 \underset{K}{\overset{*}{\to}} u_1 \underset{\Sigma,J}{\twoheadrightarrow} u_2 \underset{K}{\overset{*}{\to}} u_3 \underset{\Sigma,J}{\twoheadrightarrow} u_4 \underset{K}{\overset{*}{\to}} \ldots u_{2n-1} \underset{\Sigma,J}{\twoheadrightarrow} u_{2n} \underset{K}{\overset{*}{\to}} u_{2n+1}.$$

Soit :

$$w_0 \in M(H \cup \Phi, D_J \cup \{y_1, \ldots, y_m\}) \qquad \text{et} \qquad d_1, \ldots, d_m$$

tels que :

$$u_0 = w_0[d_1, \ldots, d_m].$$

On va définir un calcul de la forme :

$$w_0 = w_1 \underset{\Sigma,J}{\twoheadrightarrow} w_3 \underset{\Sigma,J}{\twoheadrightarrow} w_5 \underset{\Sigma,J}{\twoheadrightarrow} \ldots \underset{\Sigma,J}{\twoheadrightarrow} w_{2n-1} \underset{\Sigma,J}{\twoheadrightarrow} w_{2n+1}.$$

tel que :

$$w_{2i-1}[d_1, \ldots, d_m] \underset{K}{\overset{*}{\to}} u_{2i-1} \quad \text{pour tout } i = 1, 2, \ldots, n.$$

On utilisera le lemme suivant :

(4.3.3) LEMME : *Soit* $w \in M(H \cup \Phi, D_J \cup \{y_1, \ldots, y_m\})$, $d_1, \ldots, d_m \in D_K$ *de sortes correspondant à* $y_1, \ldots, y_m$; *soit* u *tel que* $w[d_1, \ldots, d_m] \underset{K}{\overset{*}{\to}} u \underset{\Sigma,J}{\twoheadrightarrow} u'$.

Alors, il existe w' *tel que* $w \underset{\Sigma,J}{\twoheadrightarrow} w'$ *et* $w'[d_1, \ldots, d_m] \underset{K}{\overset{*}{\to}} u'$.

Preuve : Par induction sur la structure de w.

Cas de base : $w \in D_J \cup \{y_1, \ldots, y_m\} \cup H_0$;

Ce cas ne peut pas se produire car u doit contenir au moins une occurrence d'un symbole de Φ.

Induction :

1^er^ *cas :*

$$w = \varphi(d'_1, \ldots, d'_h; w_1, \ldots, w_l),$$
$$u = \varphi(d'_1, \ldots, d'_h; u_1, \ldots, u_l)$$

avec :

$$w_i[d_1, \ldots, d_m] \xrightarrow[K]{*} u_i \quad \text{pour tout } i = 1, \ldots, l,$$

et l'occurrence de φ mise en évidence est réécrite par $u \underset{\Sigma,J}{\longrightarrow\!\!\!\!\!\rightarrow} u'$.

Alors :

$$u' = t[d''_1, \ldots, d''_h, u_1/y_1, \ldots, u_l/y_l].$$

Il suffit de poser :

$$w' = t[d''_1, \ldots, d''_h, w_1/y_1, \ldots, w_l/y_l]$$

et l'on a bien :

$$w \underset{\Sigma,J}{\longrightarrow\!\!\!\!\!\rightarrow} w' \qquad \text{et} \qquad w'[d_1, \ldots, d_m] \xrightarrow[K]{*} u'.$$

2^e^ *cas :*

w s'écrit de même et l'un des $w_1, \ldots, w_l$, disons w_1, est réécrit par $\underset{\Sigma, J}{\longrightarrow\!\!\!\!\!\rightarrow}$.

Autrement dit :

$$w_1[d_1, \ldots, d_m] \xrightarrow[K]{*} u_1 \underset{\Sigma,J}{\longrightarrow\!\!\!\!\!\rightarrow} u'_1$$

et :

$$w_i[d_1, \ldots, d_m] \xrightarrow[K]{*} u_i \quad \text{pour tout } i = 2, \ldots, l.$$

Par hypothèse d'induction, on a l'existence de w'_1 tel que :

$$w_1 \underset{\Sigma,J}{\longrightarrow\!\!\!\!\!\rightarrow} w'_1 \qquad \text{et} \qquad w'_1[d_1, \ldots, d_m] \xrightarrow[K]{*} u'_1.$$

On définit alors w' par :

$$w' = \varphi\,(d'_1, \ldots, d'_h;\ w'_1, w_2, \ldots, w_l).$$

3e *cas :*

$$w = f(w_1, \ldots, w_l) \qquad \text{et} \qquad f \in H.$$

Ce cas est très semblable au précédent. On suppose que :

$$w_1[d_1, \ldots, d_m] \underset{K}{\overset{*}{\to}} u_1 \underset{\Sigma,J}{\twoheadrightarrow} u'_1,\ w'_1[d_1, \ldots, d_n] \underset{K}{\overset{*}{\to}} u_i$$

pour $i = 2 \ldots l$, et l'on prend :

$$w' = f(w'_1, w_2, w_3, \ldots, w_l). \quad \square$$

4.3.4 *Définition de la trace*

Soit donc :

$$u_0 \underset{K}{\overset{*}{\to}} u_1 \underset{\Sigma,J}{\twoheadrightarrow} u_2 \underset{K}{\overset{*}{\to}} u_3 \underset{\Sigma,J}{\twoheadrightarrow} \ldots \underset{\Sigma,J}{\twoheadrightarrow} u_{2n} \underset{K}{\overset{*}{\to}} u_{2n+1},$$

et :

$$w_0 \in M(H \cup \Phi,\ D_J \cup \{y_1, \ldots, y_m\}) \qquad \text{tel que} \quad u_0 = w_0[d_1, \ldots, d_m].$$

On définit :

$$w_1 = w_0 \quad \text{et donc} \quad w_1[d_1, \ldots, d_m] \underset{K}{\overset{*}{\to}} u_1.$$

Par application du lemme (4.3.3), on définit w_3 tel que :

$$w_1 \underset{\Sigma,J}{\twoheadrightarrow} w_3 \qquad \text{et} \qquad w_3[d_1, \ldots, d_m] \underset{K}{\overset{*}{\to}} u_2$$

et donc :

$$w_3[d_1, \ldots, d_m] \underset{K}{\overset{*}{\to}} u_3.$$

On définit alors w_5 en utilisant (4.3.3).

De manière générale, ayant défini w_{2i-1} tel que :

$$w_{2i-1}[d_1, \ldots, d_m] \underset{K}{\overset{*}{\to}} u_{2i-1},$$

le lemme (4.3.3) définit w_{2i+1} tel que :

$$w_{2i-1} \underset{\Sigma,J}{\twoheadrightarrow} w_{2i+1}, \qquad w_{2i+1}\,[d_1, \ldots, d_m] \overset{*}{\underset{K}{\to}} u_{2i};$$

et donc :

$$w_{2i+1}\,[d_1, \ldots, d_m] \overset{*}{\underset{K}{\to}} u_{2i+1}$$

puisque :

$$u_{2i} \overset{*}{\underset{K}{\to}} u_{2i+1}.$$

Cette construction est représentée sur la figure 1, où les parties hachurées correspondent à l'utilisation du lemme (4.3.3). □

La trace d'un calcul est un calcul dans une interprétation particulière, « libre » en ce sens que les opérations définies par les fonctions $(f_I)_{f \in H}$ ne sont pas effectuées.

Cette interprétation est l'interprétation *semi-libre* associée à I, notée $L(I)$. On la notera aussi $L(J)$ car elle ne dépend que de la semi-interprétation J sous-jacente à I.

L'interprétation $L(I)$ est l'interprétation $I' = \langle (E_s)_{s \in \mathscr{S}}, (f_{I'})_{f \in F} \rangle$ définie comme suit :

$Y = (Y_s)_{s \in \mathscr{U}}$ est un ensemble de variables (chaque Y_s est dénombrable);

$$E_s = D_s \qquad \text{si} \quad s \in \mathscr{T}$$

$$E_s = M(H, D_J \cup Y)_s \qquad \text{si} \quad s \in \mathscr{U},$$

$$f_{I'} = f_I (= f_J) \qquad \text{si} \quad f \in G,$$

$$f_{I'} = \lambda e_1, \ldots, e_n . f(e_1, \ldots, e_n)$$

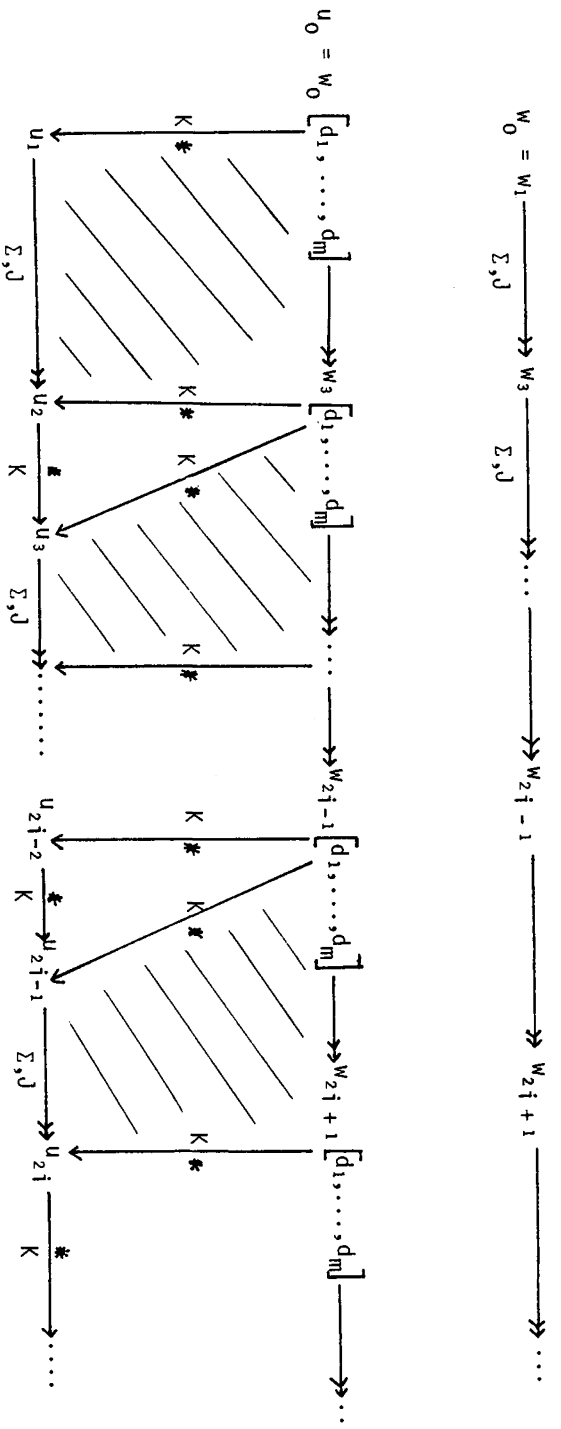
si :

$$f \in H_n (\text{i.e. } f \in H \text{ et } \rho(f) = n)$$

(et les sortes de $e_1, \ldots, e_n$ sont compatibles avec l'arité de f).

4.4. Propriétés d'un système semi-interprété; théorème fondamental

On a déjà noté que le fait pour un système interprété (Σ, I) d'être déterministe (resp. partout défini) ne dépend que des *membres gauches* des règles de Σ et des fonctions f_I pour chaque fonction de base f y apparaissant. Il en résulte [compte tenu de (4.1.1)] que ces définitions s'appliquent immédiatement au système semi-interprété sous-jacent à un système interprété donné.


$u_0 = w_0$
$w_0 = w_1$
Σ, J
K
$*$
$[d_1, \ldots, d_m]$
u_1
u_2
u_3
u_{2i-2}
u_{2i-1}
u_{2i}
w_3
w_{2i-1}
w_{2i+1}
w_{2i-1}
w_{2i+1}

On distinguera donc les systèmes semi-interprétés *déterministes* (resp. *partout définis*).

Un système semi-interprété (Σ, J) est *cohérent* (resp. *confluent, Noetherien, total*) si, pour toute interprétation $I = J + K$ (i.e. ayant J pour semi-interprétation sous-jacente), le système (Σ, I) est cohérent (resp. confluent, Noetherien, total).

REMARQUE : Les énoncés des propositions (1.6.1) et (1.7.1) s'appliquent immédiatement aux systèmes semi-interprétés.

Notre but va être de comparer un système semi-interprété (Σ, J) avec le système interprété $(\Sigma, L(J))$.

Nous allons démontrer le résultat fondamental suivant :

4.4.1. THÉORÈME : *Un système semi-interprété* (Σ, J) *est cohérent (resp. confluent) si et seulement si le système interprété* $(\Sigma, L(J))$ *est cohérent (resp. confluent).*

Preuve :

1° La condition est nécessaire puisque $L(J)$ est une interprétation dont la semi-interprétation sous-jacente est J.

2° *Réciproquement :*

(*a*) *cohérence :* supposons $(\Sigma, L(J))$ cohérent. Soit I une interprétation de la forme $J + K$ et :

$$u_0 = w_0[d_1, \ldots, d_m] \Rightarrow u_1 \Rightarrow u_2 \ldots \Rightarrow u_k \Rightarrow d,$$

un calcul de $t[d_1, \ldots, d_m]$ où $\Rightarrow$ est la relation $\underset{\Sigma, I}{\Rightarrow}$.

Ce calcul a une *trace* :

$$w_0 = w_1 \twoheadrightarrow w_2 \twoheadrightarrow w_3 \twoheadrightarrow \ldots \twoheadrightarrow w_l = w,$$

dans $L(J)$ (où $\twoheadrightarrow$ est la relation $\underset{\Sigma, J}{\twoheadrightarrow}$), telle que :

$$w \in M(H, D_J \cup Y_m) \qquad \text{et} \qquad d = w_I(d_1, \ldots, d_m).$$

Si par ailleurs on avait $u_0 \overset{*}{\Rightarrow} d'$, ce calcul aurait une trace $w_0 \overset{*}{\twoheadrightarrow} w'$ telle que $d' = w'_I(d_1, \ldots, d_m)$.

Puisque w, w' appartiennent à $M(H, D_J \cup Y_m)$ (s'ils contenaient des occurrences de symboles de fonctions appartenant à Φ, ils n'auraient pas de valeurs dans I car les fonctions de base sont supposées strictes) et que $(\Sigma, L(J))$ est cohérent, $w = w'$ et donc $d = d'$.

(*b*) *confluence :*

Soit $(\Sigma, L(J))$ confluent et $I = J + K$.

Soit $t \overset{*}{\Rightarrow} t_1$ et $t \overset{*}{\Rightarrow} t_2$ deux calculs de (Σ, I).

Le lemme (4.3.3) permet de construire les traces :

$$w \overset{*}{\twoheadrightarrow} w_1 \qquad \text{et} \qquad w \overset{*}{\twoheadrightarrow} w_2$$

de ces calculs, avec :

$$w[d_1, \ldots, d_m] \underset{K}{\overset{*}{\rightarrow}} t,$$

$$w_i[d_1, \ldots, d_m] \underset{K}{\overset{*}{\rightarrow}} t_i \qquad (\text{pour } i = 1, 2).$$

On notera désormais $\bar{w}$, $\bar{w}_1$, $\bar{w}_2$ au lieu de $w[d_1, \ldots, d_m]$, etc.

La confluence de $(\Sigma, L(J))$ donne l'existence de w' :

$$w_i \overset{*}{\twoheadrightarrow} w',$$

d'où il résulte :

$$\bar{w}_i \overset{*}{\twoheadrightarrow} \bar{w}'$$

et l'existence de $u_i \in M(H \cup \Phi, D)$ tel que :

$$t_i \overset{*}{\twoheadrightarrow} u_i \qquad \text{et} \qquad \bar{w}' \underset{K}{\overset{*}{\rightarrow}} u_i.$$

[par un lemme semblable à (4.3.3)].

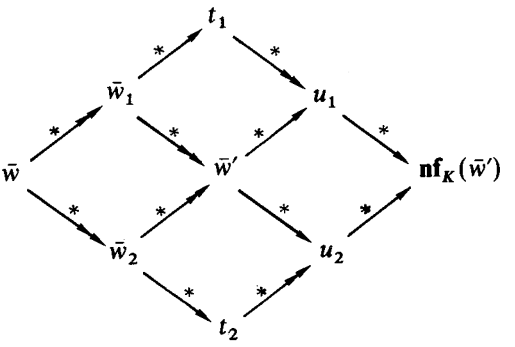
On a alors :

$$\bar{w}' \underset{K}{\overset{*}{\rightarrow}} u_i \underset{K}{\overset{*}{\rightarrow}} \mathbf{nf}_K(\bar{w}')$$

et donc :

$$t_i \overset{*}{\Rightarrow} \mathbf{nf}_K(\bar{w}') \qquad \text{pour} \quad i = 1, 2.$$

Cette preuve peut être résumée par le diagramme :



où : $\overset{*}{\twoheadrightarrow}$ est la relation $\underset{\Sigma, J}{\overset{*}{\twoheadrightarrow}}$, $\overset{*}{\rightarrow}$ est la relation $\underset{K}{\overset{*}{\rightarrow}}$.

4.4.2. THÉORÈME : *Un système semi-interprété* (Σ, J) *est total (resp. Noetherien) si et seulement si il existe une interprétation* $I = J + K$ *telle que :* (Σ, I) *soit total (resp. Noetherien) si et seulement si* $(\Sigma, L(J))$ *est total (resp. Noetherien).*

Preuve : Tout revient à montrer que, si $I = J + K$, alors (Σ, I) est total (resp. Noetherien) si et seulement si $(\Sigma, L(J))$ est total (resp. Noetherien). Cela se fait facilement au moyen de la notion de trace d'un calcul. □

4.5. Applications

La notion de système semi-interprété et les théorèmes (4.4.1) et (4.4.2) permettent de prouver le déterminisme, la confluence, la cohérence, la totalité, etc. de tous les systèmes interprétés ayant un même système semi-interprété sous-jacent en *une seule fois*, et pour le système semi-interprété en question.

Nous allons maintenant préciser les questions de déterminisme et de confluence, pour les systèmes semi-interprétés (Σ, J) où $J = M(G)$, ce qui constitue un cas extrêmement fréquent comme le montrent les exemples (2.1), (2.5), (2.6), (2.7) et (2.8).

4.5.1. LEMME : $(\Sigma, M(G))$ *est déterministe si et seulement si* Σ *est sans paires critiques.*

Preuve : Pour la définition de la notion de paire critique, on se reportera à Huet [14].

Soit Σ un système spécial, (s_1, t_1) et (s_2, t_2) deux couples appartenant à Σ. Puisque s_1 et s_2 sont dans $\Phi(M(G, V))$, le seul sous-terme de s_1 qui puisse être unifiable avec s_2 est s_1 lui-même.

Supposons que (s_1, t_1) et (s_2, t_2) engendrent une paire critique $\langle u_1, u_2 \rangle$ et donc un terme critique u de $\Phi(M(G, V))$ tel que :

$$u \underset{\Sigma, M(G)}{\twoheadrightarrow} u_1 \quad \text{et} \quad u \underset{\Sigma, M(G)}{\twoheadrightarrow} u_2$$

(u se déduit de s_1 par une substitution).

u est de la forme $u = \varphi(v_1, \ldots, v_n; y_1, \ldots, y_m)$ avec :

$$\varphi \in \Phi,$$

$$v_1, \ldots, v_n \in M(G, X_k); \qquad X_k = \{x_1, \ldots, x_k\}.$$

Si $\alpha_1, \ldots, \alpha_k$ désignent des constantes de G, le terme :

$$u[\alpha_1/x_1, \alpha_2/x_2, \ldots, \alpha_k/x_k]$$

est tel que les deux règles (s_1, t_1) et (s_2, t_2) s'appliquent à lui.

Donc $(\Sigma, M(G))$ n'est pas déterministe.

Réciproquement, supposons que $(\Sigma, M(G))$ ne soit pas déterministe.

Il existe alors $\varphi \in \Phi$ et un l-uplet $(d_1, \ldots, d_n; y_1, \ldots, y_m)$ — où $d_1, \ldots, d_n \in M(G)$ — tels que :

Il existe au moins deux règles (s_1, t_1) et (s_2, t_2) s'appliquant au terme $u = \varphi(d_1, \ldots, d_n; y_1, \ldots, y_m)$.

En effet, la deuxième condition contenue dans la définition du déterminisme (1.6) est toujours satisfaite lorsque l'interprétation a lieu dans $M(G)$, puisqu'on s'intéresse à l'égalité formelle des termes. Alors : u est le terme critique associé à la paire critique engendrée par le couple $\langle (s_1, t_1), (s_2, t_2) \rangle$. □

u est le terme critique associé à la paire critique engendrée par le couple $\langle (s_1, t_1), (s_2, t_2) \rangle$. □

4.5.2. *Application* : Le lemme (4.5.1) fournit une *condition nécessaire et suffisante décidable de déterminisme* pour le système semi-interprété $(\Sigma, M(G))$. En effet, il n'y a qu'un nombre fini de couples de paires de règles de Σ à examiner.

4.5.3. PROPOSITION : *Un n-uplet $(s_1, \ldots, s_n)$ d'éléments de $\Phi(M(G, V))$ est sans paires critiques si et seulement si pour tous $t_1, \ldots, t_n$ tels que le système $(\Sigma, M(G))$, où $\Sigma = \{(s_i, t_i)/1 \leqq i \leqq n\}$ soit Noetherien, $(\Sigma, M(G))$ est confluent.*

Preuve : $(\Sigma, M(G))$ étant Noetherien, on peut lui appliquer le théorème 2 de Huet [14].

Supposons que $(s_1, \ldots, s_n)$ n'admette pas de paire critique; alors la condition nécessaire et suffisante de confluence (égalité des formes normales pour les deux composantes de toute paire critique) est satisfaite, donc $(\Sigma, M(G))$ est confluent.

Réciproquement, supposons qu'il existe une paire critique déterminée par s_i et s_j $(1 \leqq i \leqq n;\ 1 \leqq j \leqq n;\ i < j)$.

Soit u le terme critique associé. On peut alors trouver un n-uplet $(t_1, \ldots, t_n)$ tel que $(\Sigma, M(G))$ soit Noetherien et non-confluent.

Il suffit de choisir les règles :

$$\left.\begin{array}{c} s_{i'} = a \quad \text{pour tout } i' \neq j. \\ s_j = b \text{ (où } a \text{ et } b \text{ désignent des constantes distinctes de } H). \end{array}\right\}$$

Et l'on a alors :

$$u \underset{\Sigma,\ (M(G))}{\longrightarrow\!\!\!\!\!\rightarrow} a \qquad \text{et} \qquad u \underset{\Sigma,\ M(G)}{\longrightarrow\!\!\!\!\!\rightarrow} b,$$

ce qui prouve que $(\Sigma, M(G))$ n'est pas confluent. □

RÉFÉRENCES

1. S. Bloom et R. Tindell, *Varieties of if Then-else*, Siam J. Comput. vol. 12, 1983, p. 677-707.
2. G. Boudol, *Calculs maximaux et sémantique opérationnelle des programmes non déterministes*, Thèse doctorat, Université de Paris-7, 1980.
3. B. Courcelle, *Infinite Trees in Normal Form and Recursive Equations having a Unique Solution*, Math. Systems Theory, vol. 13, 1979, p. 131-180.
4. B. Courcelle, *Fundamental Properties of Infinite Trees*, Theor. Comput. Sc., vol. 25, 1984, p. 95-169.
5. B. Courcelle, *An Axiomatic Approach to the Korenjak-Hopcroft Algorithms*, Math. Systems Theory, vol. 16, 1983, p. 191-231.
6. B. Courcelle et P. Franchi-Zannettacci, Attribute Grammars and Recursive Program Schemes, Theor. Comput. Sc., vol. 17, 1982, p. 163-191 and p. 235-257.
7. B. Courcelle et F. Lavandier, *A Class of Program Schemes Based on Tree Rewriting Systems 8th C.A.A.P.*, L'Aquila, Lec. Notes Comp. Sc., Springer, vol. 159, 1983, p. 191-204.
8. B. Courcelle, M. Nivat, *The algebraic semantics of recursive program schemes: MFCS'78*, Lec. Notes Comput. Sc., vol. 64, p. 16-30.
9. J. Darlington, R. Burstall, *A System which Automatically Improves Programs*, Acta Informatica, vol. 6, 1976, p. 41-60.
10. E. Friedman, *Equivalence Problems for Deterministic Context-free Languages and Monadic Recursion Schemes*, J. Comput. System Sc., 14, 1977, p. 334-359.
11. S. Garland et D. Luckham, *Program Schemes, Recursion Schemes and Formal Languages*, J. Comput. System Sc., vol. 7, 1973, p. 119-160.

12. J. GOGUEN, J. THATCHER, E. WAGNER et J. WRIGHT, *Initial Algebra Semantics and Continuous Algebras*, J. Assoc. Comput. Mach., vol. 24, 1977, p. 68-95.

13. I. GUESSARIAN, *Algebraic Semantics*, Lec. Notes Comput. Sc., vol. 99, Springer-Verlag, 1981.

14. G. HUET, *Confluent Reductions, Abstract Properties and Applications to Term Rewriting Systems*, J. Assoc. Comput. Mach., vol. 27, 1980, p. 797-821.

15. G. HUET et J. M. HULLOT, *Proofs by Induction in Equational Theories with Constructors*, J. Comput. System Sc., vol. 25, 1982, p. 239-266.

16. G. HUET et B. LANG, *Proving and Applyind Program Transformations Expressed with 2nd Order Patterns*, Acta Informatica, vol. 11, 1978, p. 31-55.

17. G. HUET et J. J. LEVY, *Call by Need Computations in Nonambigous Linear Term Rewriting Systems*, Laboria report, vol. 359, 1979.

18. G. HUET et D. OPPEN, *Equations and Rewrite Rules, a Survey*, Proceedings of the International Symposium on Formal Languages Theory, Santa Barbara, California (December 10-14, 1979), Academic Press, 1980.

19. F. LAVANDIER, *Sur les systèmes de définitions récursives par cas. Application à la sémantique dénotationnelle*, Thèse de 3e cycle, Université de Bordeaux-I, 1982.

20. Z. MANNA, A. SHAMIR, *The Theoretical Aspects of the Optimal Fixedpoint*, S.I.A.M., J. Comput., vol. 5, 1976, p. 414-426.

21. B. MAYOH, *Attribute Grammars and Mathematical Semantics;* S.I.A.M. J. Comput., vol. 10, 1981, p. 503-518.

22. M. NIVAT, *On the Interpretation of Polyadic Recursive Program Schemes*, Symposia Mathematica, vol. 15, Academic Press, 1975, p. 255-281.

23. J. C. RAOULT et J. VUILLEMIN, *Operational and Semantic Equivalence between Recursive Programs*, J. Assoc. Comput. Mach., vol. 27, 1980, p. 772-796.

24. B. ROSEN, *Tree Manipulation Systems and Church-Rosser Theorems*, J. Assoc. Comput. Mach., vol. 20, 1973, p. 160-187.

25. J. STOY, *Semantic models*, in *Theoretical Foundations of Programming Methodology*, M. BROY and G. SCHMIDT, éd., D. Reidel Pub. Co., Dordrecht, Holland, 1982, p. 293-325.

26. R. TENNENT, *The denotational Semantics of Programming Languages*, Comm. of A.C.M., vol. 19-8, 1976, p. 437-453.

27. S. WALKER et H. STRONG, *Characterizations of Flow-chartable Recursions*, J. Comput. *System Sc.*, vol. 7, 1973, p. 404-447.

28. J. ENGELFRIET, *Some Open Questions and Recent Results on Tree Transducers and Tree Languages*, même volume que [18].

29. P. FRANCHI-ZANNETTACCI, *Attributs sémantiques et schémas de programmes*, Thèse d'État, Université de Bordeaux-I, 1982.