

81-164

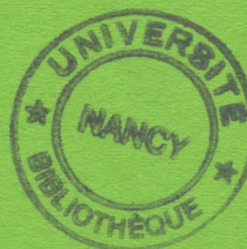
UNIVERSITÉ DE NANCY I

CENTRE DE RECHERCHE EN
INFORMATIQUE DE NANCY

Sc N 81 / 134 B

LES ORDRES DE DÉCOMPOSITION :
UN OUTIL INCRÉMENTAL POUR PROUVER
LA TERMINAISON FINIE DE SYSTÈMES
DE RÉÉCRITURE DE TERMES

THESE



présentée pour l'obtention du grade de
DOCTEUR DE 3^e CYCLE EN INFORMATIQUE

PAR

FERNAND REINIG

soutenue publiquement le 30 Octobre 1981

JURY

Président M. C. PAIR
Examineurs M. J.P. FINANCE
M. J.P. JOUANNAUD
P. LESCANNE
J.J. LEVY

UNIVERSITÉ DE NANCY I

CENTRE DE RECHERCHE EN
INFORMATIQUE DE NANCY

LES ORDRES DE DÉCOMPOSITION :
UN OUTIL INCRÉMENTAL POUR PROUVER
LA TERMINAISON FINIE DE SYSTÈMES
DE RÉÉCRITURE DE TERMES



THESE

présentée pour l'obtention du grade de
DOCTEUR DE 3^e CYCLE EN INFORMATIQUE

PAR

FERNAND REINIG

soutenue publiquement le 30 Octobre 1981

JURY

Président M. C. PAIR
Examineurs M. J.P. FINANCE
 M. J.P. JOUANNAUD
 P. LESCANNE
 J.J. LEVY

Monsieur le professeur Claude PAIR m'a initié à la recherche en informatique dans un cours de DEA où j'ai beaucoup admiré la précision et la clarté avec lesquelles il a su présenter les bases théoriques de la programmation. Malgré ses nouvelles fonctions à Paris et son emploi du temps chargé, il a accepté de présider mon jury. Je lui adresse ici mes sincères remerciements.

Jean-Pierre JOUANNAUD a dirigé mon travail de thèse. Par ses suggestions et ses critiques constructives, il m'a permis de clarifier peu à peu les concepts essentiels de ce travail. Il m'a conseillé judicieusement jusqu'à la rédaction finale de cette thèse et je lui adresse ici ma profonde gratitude.

Pierre LESCANNE m'a poussé à m'intéresser au problème des ordres de décomposition. Il a toujours suivi avec un vif intérêt l'avancement de mes travaux, de près d'abord, de loin ensuite après être parti au MIT. Je lui adresse ici tous mes remerciements.

Jean-Pierre FINANCE m'a enseigné les bases de l'informatique théorique. Il a bien voulu participer à mon jury et je l'en remercie.

Je remercie Jean-Jacques LEVY pour l'intérêt qu'il a porté à mon travail et pour la gentillesse qu'il a manifestée quand j'ai eu besoin de ses conseils.

Finalement j'adresse mes remerciements à Madame DRIQUERT pour avoir accepté la tâche ingrate de taper cette thèse et je la félicite de son très beau travail.

TABLE DES MATIÈRES

<u>INTRODUCTION</u>	1
<u>CHAPITRE I : NOTIONS PRELIMINAIRES</u>	4
1. <u>MULTI-ENSEMBLES</u>	5
1.1. Définitions	5
1.2. Ordre de Manna-Dershowitz	6
1.3. Partition sur les multi-ensembles	9
1.4. Ordre basé sur les partitions	12
2. <u>TERMES</u>	15
3. <u>SYSTEMES DE REECRITURE ET ORDRES DE SIMPLIFICATION</u>	18
<u>CHAPITRE II : ORDRES DE SIMPLIFICATION RECURSIFS</u>	22
1. <u>PLONGEMENT</u>	23
2. <u>ORDRE DE KNUTH-BENDIX</u>	28
3. <u>ORDRE RECURSIF SUR LES CHEMINS</u>	33
4. <u>ORDRE LEXICOGRAPHIQUE SUR LES CHEMINS</u>	37
5. <u>ORDRE RECURSIF GENERALISE SUR LES CHEMINS</u>	38
6. <u>ORDRE DE PLAISTED</u>	41
7. <u>CONCLUSION</u>	41
<u>CHAPITRE III : ORDRES DE DECOMPOSITION</u>	43
1. <u>DEFINITION DE L'ORDRE DE DECOMPOSITION</u>	44
2. <u>LES APPLICATIONS DE CHOIX DANS L'ORDRE DE DECOMPOSITION</u>	53
2.1. Le choix complet et le choix maximal	53
2.2. La dépendance de l'ordre de décomposition des applications de choix ..	58

3. <u>L'ORDRE DE DECOMPOSITION EST UN ORDRE DE SIMPLIFICATION</u>	69
4. <u>L'ORDRE DE DECOMPOSITION EST UN ORDRE BIEN FONDE</u>	76
5. <u>COMPARAISON ENTRE L'ORDRE DE DECOMPOSITION ET L'ORDRE RECURSIF SUR LES CHEMINS</u> ..	83
6. <u>EXTENSION DE L'ORDRE DE DECOMPOSITION AUX TERMES NON CLOS</u>	90
<u>CONCLUSION</u>	97
<u>BIBLIOGRAPHIE</u>	103

INTRODUCTION

Le théorie des systèmes de réécriture de termes est d'une grande importance pratique en programmation (sémantique opérationnelle, démonstration automatique, spécification de types abstraits algébriques,...).

Un système de réécriture est un ensemble d'équations orientées appelées règles. Réécrire un terme signifie remplacer un sous-terme de ce terme qui est une instance du membre gauche d'une règle par l'instance correspondante du membre droit de cette règle. Le choix du sous-terme et de la règle est indéterminée.

Parmi les nombreux problèmes théoriques que posent ce modèle de calcul, celui de la terminaison finie est d'une grande importance. On sait que la terminaison d'un système de réécriture avec au moins deux règles est indécidable [HL 78]. On est ainsi amené à trouver des méthodes de preuve de terminaison finie à l'aide de conditions suffisantes.

Itturiaga [ITT 67] d'une part, Lipton et Snyder [LS 77] d'autre part ont donné des conditions suffisantes de terminaison pour certaines classes de systèmes de réécriture.

La méthode de Manna et Ness [MN 70], [LAN 75] est basée sur l'existence d'un ordre noethérien compatible avec la relation de réécriture. Pour pouvoir appliquer cette méthode, il faut trouver pour chaque système de réécriture un ordre noethérien adapté aux règles : Gorn [GOR 67], Knuth et Bendix [KB 70], Plaisted [PLA 78], Dershowitz et Manna [DM 79] ont donné différents ordres bien fondés sur différents ensembles, comme l'ordre sur les chemins des sous-termes, l'ordre multi-ensemble,...

La méthode proposée par Dershowitz [DER 79] est basée sur la notion d'ordre de simplification. Les conditions imposées à un ordre pour être un ordre de simplification sont relativement faciles à vérifier et parmi les ordres noethériens proposés antérieurement, il y avait de nombreux ordres de simplification [KB 70], [PLA 78].

De nouveaux ordres de simplification définis récursivement ont été proposés ensuite par Dershowitz [DER 79], Kamin et Lévy [KL 80] et Lescanne [LES 81]. L'ordre de décomposition qui est étudié en détail dans cette thèse, est une généralisation de l'ordre défini par P. Lescanne. Il s'agit d'un ordre de simplification défini récursivement à partir d'un ordre de base sur les symboles de l'algèbre et qui possède quelques propriétés classiques telle que la monotonie pour l'extension de l'ordre sur les symboles et la clôture par instanciation des variables.

Un inconvénient pratique important concernant les ordres de simplification récursifs est que l'ordre partiel $<$ sur les symboles de fonction de l'algèbre doit être donné a priori. Si cet ordre ne convient pas, il faut recommencer avec un autre. D'où l'idée de construire l'ordre partiel $<$ au fur et à mesure des besoins. Cela est théoriquement valide grâce à la propriété de monotonie : la croissance de l'ordre $<$ ne remet pas en cause les calculs réussis antérieurement. Cela ne sera possible en pratique que si l'échec de l'orientation de deux termes s et t fournit les indications nécessaires à la construction du nouvel ordre $<$ contenant $<$ et permettant d'orienter la paire (s, t) à l'aide de l'ordre de simplification récursif. Nous appelons cette propriété : "propriété d'incrémentalité" d'un ordre de simplification récursif.

Pour diverses raisons, les ordres de simplification définis par Knuth et Bendix [KB 70], par Plaisted [PLA 78], par Dershowitz [DER 79], par Lescanne [LES 81] n'ont pas la propriété d'incrémentalité. Nous affirmons (sans le prouver) que l'ordre de décomposition a la propriété d'incrémentalité et qu'il est donc mieux adapté que les autres à l'orientation automatique d'équations. Afin d'en convaincre le lecteur nous étudions quelques cas pratiques et nous donnons la raison essentielle de cette propriété.

Après avoir introduit dans le chapitre I les notions indispensables sur les multi-ensembles, les termes, les systèmes de

réécriture et les ordres de simplification, nous faisons dans le chapitre II une étude succincte des principaux ordres de simplification récursifs. Dans le chapitre III nous donnons la définition de l'ordre de décomposition, nous étudions en détail les différentes propriétés de cet ordre et nous prouvons qu'il est strictement plus fin que l'ordre récursif sur les chemins. Finalement, nous illustrons à l'aide de trois exemples présentés en conclusion la propriété d'incrémentalité de l'ordre de décomposition.

CHAPITRE 1 :

NOTIONS PRÉLIMINAIRES

1.- MULTI-ENSEMBLES

1.1.- Définitions

Un multi-ensemble d'éléments de A est une collection non ordonnée d'éléments de A où chaque élément peut apparaître plusieurs fois. Plus précisément, on a la définition suivante (l'ensemble des entiers positifs étant noté $\mathbb{N}$) :

Définition I.1 :

Un multi-ensemble sur l'ensemble A est une application $M : A \rightarrow \mathbb{N}$.

Ainsi $M = \{1, 3, 5, 5, 3, 5\}$ est un multi-ensemble sur $\mathbb{N}$
avec $M(1) = 1$
 $M(3) = 2$
 $M(5) = 3$
et $M(x) = 0$ pour tout autre x dans $\mathbb{N}$.

On note par $\mathcal{M}(A)$ l'ensemble de tous les multi-ensembles finis dont les éléments appartiennent à A , c'est-à-dire toutes les applications $M : A \rightarrow \mathbb{N}$ dont le support $\{x \in A \mid M(x) \neq 0\}$ est fini.

Le multi-ensemble noté $\{\}$ tel que $\{\}(x) = 0$ pour tout x dans A s'appelle le multi-ensemble vide. On dit qu'un élément x de A appartient à M et on note $x \in M$ ssi $M(x) > 0$.

Donnons maintenant la définition de l'union et de la différence de deux multi-ensembles.

Définition I.2 :

L'union $M_1 + M_2$ de deux multi-ensembles M_1 et M_2 est dé-

finie par :

$$M_1 + M_2(x) = M_1(x) + M_2(x) \text{ pour tout } x \text{ dans } A.$$

Cette union restreinte aux ensembles coïncide avec l'union disjointe ou la somme directe.

La différence $M_1 - M_2$ de deux multi-ensembles n'est définie que si M_2 est inclus dans M_1 .

Définition I.3 :

On dit que le multi-ensemble M_2 est inclus dans le multi-ensemble M_1 si

$$M_2(x) \leq M_1(x) \text{ pour tout } x \text{ dans } A.$$

Définition I.4 :

Si M_2 est inclus dans M_1 , alors la différence $M_1 - M_2$ des multi-ensembles M_1 et M_2 est définie par :

$$M_1 - M_2(x) = M_1(x) - M_2(x) \text{ pour tout } x \text{ dans } A.$$

1.2.- Ordre de Manna-Dershowitz sur les multi-ensembles

Un ordre partiel strict $<$ sur un ensemble A est une relation binaire irréflexive et transitive. Nous allons introduire des extensions de l'ordre $<$ sur A à l'ensemble $\mathcal{M}(A)$. Ces extensions seront notées $<<$.

Si deux éléments x et y de A sont incomparables pour l'ordre $<$, nous notons : $x \# y$.

Donnons maintenant la définition de l'extension introduite par Manna et Dershowitz [DM 79] :

Définition I.5 :

Soient M et M' deux éléments de $\mathcal{M}(A)$. On dit que :

$$M << M' \text{ ssi}$$

il existe deux éléments X et Y de $\mathcal{M}(A)$, avec $X \neq \{\}$ et $X \subseteq M'$, tels que :

$$M = (M' - X) + Y$$

$$\text{et } (\forall y \in Y) (\exists x \in X) x > y.$$

On voit donc qu'on réduit un multi-ensemble en enlevant un ou plusieurs éléments et en les remplaçant par des multi-ensembles d'éléments plus petits.

Exemple :

$$\text{Si } A = \{1, 2, 3, \dots\} \cup \{a, b, c, \dots\}$$

$$\text{avec } 1 < 2 < 3 \dots$$

$$\text{et } a < b < c \dots$$

on a :

$$\{2, 2, 2, 3, 4, a, a, b, c, c\} << \{2, 3, 3, 4, b, b, d\}$$

car on a :

$$X = \{d, b, 3\}$$

$$\text{et } Y = \{2, 2, a, a, c, c\}.$$

Une autre définition de $<<$ est donnée par Huet et Oppen [HO 80] :

Définition I.6 :

Soient M et M' deux éléments de $\mathcal{M}(A)$. On dit que :

$$M \ll M' \text{ ssi}$$

$$M \neq M'$$

$$\text{et } (\forall x \in A)(M'(x) < M(x)) \Rightarrow [(\exists y \in A)(x < y) \text{ et } (M(y) < M'(y))].$$

Il est aisé de voir que les définitions I.5 et I.6 sont équivalentes [JL 81].

Dans [DM 79] Manna et Dershowitz montrent que la relation $\ll$ ainsi définie est bien un ordre partiel struct sur $\mathcal{M}(A)$, si $<$ est un ordre partiel strict sur A .

Remarque : Si l'ordre $>$ sur A est total, on peut comparer deux multi-ensembles en triant d'abord leurs éléments dans l'ordre décroissant pour $>$ et en comparant ensuite lexicographiquement les listes ainsi obtenues.

Exemple :

$$\{2,3,4,2,2,4,1\} \ll \{4,3,2,3,4\}$$

$$\text{parce que : } (4,4,3,2,2,2,1) <_{\text{lexico}} (4,4,3,3,2).$$

Définition I.7 :

On dit qu'une relation R dans un ensemble A est noethérienne s'il n'existe pas de chemin infini dans la relation i.e. il n'existe pas $x_1, x_2, x_3, \dots$ dans A avec

$$x_1 R x_2 R x_3 \dots$$

On dit alors que (A, R) est bien-fondé.

On a le théorème important suivant sur l'extension de l'ordre $<$ aux multi-ensembles :

Théorème I.1 : [DM 79]

L'ensemble $(\mathcal{M}(A), \gg)$ est bien fondé si et seulement si l'ensemble $(A, >)$ est bien fondé.

Finalement on peut étendre naturellement la notion de multi-ensemble à des multi-ensembles emboîtés ; l'ordre sur ces multi-ensembles est obtenu récursivement à partir de $\ll$ [DM 79].

1.3.- Partitions sur les multi-ensembles

Nous avons vu que, lorsque l'ordre $>$ sur A est total, on peut représenter chaque multi-ensemble par une liste triée par ordre décroissant. Essayons de généraliser cette représentation dans le cas où l'ordre sur A est seulement partiel, afin d'obtenir une mise en oeuvre simple de l'ordre sur les multi-ensembles.

Pour cela introduisons la notion de partition d'un multi-ensemble.

Définition I.8 :

Les multi-ensembles $(M_1, \dots, M_n)$ forment une partition du multi-ensemble M ssi

$$(i) \quad \sum_{i=1}^n M_i = M$$

$$(ii) \quad M_i \neq \{\} \text{ pour tout } i \text{ de } 1 \text{ à } n.$$

Exemple :

Les multi-ensembles $\{3\}, \{1,5,6\}$ et $\{3,5\}$ forment une par-

tition du multi-ensemble $\{1,5,5,5,3,3\}$.

Remarquons que la restriction de cette définition à des ensembles coïncide avec la notion bien connue de "partition d'un ensemble". En effet, la condition (i) ci-dessus implique les deux conditions suivantes dans le cas des ensembles :

$$\bigcup_{i=1}^n S_i = S$$

$$\text{et } S_i \cap S_j = \emptyset \text{ pour tout } i \neq j$$

(ceci parce que $+$ coïncide avec la réunion disjointe, donc chaque élément d'un ensemble n'apparaît qu'une seule fois dans cet ensemble).

Donnons maintenant la définition d'une partition $\leftarrow$ -gauchie d'un multi-ensemble :

Définition I.9 :

La partition $(M_1, \dots, M_n)$ est une partition $\leftarrow$ -gauchie de M ssi

- (i) $M_i(x) \leq 1$ pour tout x dans A et pour tout i de 1 à n .
- (ii) pour tout x et y dans A et pour tout i de 1 à n :
 $[M_i(x) = 1 \text{ et } M_i(y) = 1] \Rightarrow (x \neq y)$.
- (iii) pour tout x dans A et pour tout i de 2 à n :
 $(M_i(x) = 1) \Rightarrow [(\exists y \in A)(M_{i-1}(y) = 1)$
et $(y = x \text{ ou } y > x)]$.

Remarquons que la condition (i) dit que pour tout i de 1 à n , M_i est un ensemble.

L'unicité de la partition $\leftarrow$ -gauchie d'un multi-ensemble est immédiate. La première classe de cette partition sera appelée par la suite la classe maximale du multi-ensemble.

Exemple :

$$M = \{4, d, d, 4, 3, 3, 2, e, 1\}$$

$$M_1 = \{4, d\} ; M_2 = \{e, d\} ; M_3 = \{3, c\} ;$$

$$M_4 = \{2, a\} ; M_5 = \{1\} ;$$

$(M_1, M_2, M_3, M_4, M_5)$ est la partition $\leftarrow$ -gauchie de M .

$M_1 = \{4, d\}$ est la classe maximale de M .

Lorsque l'ordre sur A est total, la partition $\leftarrow$ -gauchie est obtenue immédiatement à partir de la liste triée par ordre décroissant, la classe maximale étant l'ensemble contenant l'élément maximum du multi-ensemble.

Exemple :

Si $M = \{5, 4, 3, 3, 5, 2\}$, alors la liste triée correspondant à M est égale à : $\{5, 5, 4, 3, 3, 2\}$ et on obtient :

$$M_1 = \{5\} ; M_2 = \{5\} ; M_3 = \{4\} ; M_4 = \{3\} ;$$

$$M_5 = \{3\} ; M_6 = \{2\}.$$

Si l'ordre sur A n'est pas total, nous allons utiliser la notion de partition $\leftarrow$ -gauchie pour donner une représentation des multi-ensembles par des listes $\leftarrow$ -gauchies proche de celle avec des listes triées dans le cas de l'ordre total.

Définition I.10 :

Une liste $L = (e_1, \dots, e_n)$ d'éléments de A est dite <-gauchie ssi

il existe

$$\begin{array}{l} i_1 \geq 1 \\ i_2 > i_1 \\ \vdots \\ i_p = n \end{array}$$

tels que les ensembles

$$\begin{array}{l} M_1 = \{e_1, \dots, e_{i_1}\} \\ M_2 = \{e_{i_1+1}, \dots\} \\ \vdots \\ M_p = \{e_{i_{p-1}+1}, \dots, e_n\} \end{array}$$

forment la partition <-gauchie du multi-ensemble $\{e_1, \dots, e_n\}$.

Remarque : La représentation d'un multi-ensemble par une liste <-gauchie n'est pas unique.

Exemple :

$$M = \{4, d, d, c, 3, 3, 2, a, 1\}$$

Alors :

$$L_1 = (4, d, d, 3, 3, c, a, 2, 1)$$

et $L_2 = (d, 4, d, 3, c, 3, a, 2, 1)$

sont deux représentations <-gauchies du multi-ensemble M .

1.4.- Ordre basé sur les partitions

Par analogie avec la comparaison lexicographique des listes

triées dans le cas où l'ordre $>$ sur A est total, on peut comparer deux multi-ensembles en utilisant la partition <-gauchie définie ci-dessus. On notera la relation ainsi obtenue par $\ll_p$.

Définition I.11 :

Soient M et N deux multi-ensembles de $\mathcal{M}(A)$ et $(M_1, \dots, M_p)$ et $(N_1, \dots, N_q)$ leur partition <-gauchie. On dit que

$$\begin{array}{l} M \ll_p N \text{ ssi} \\ (M_1, \dots, M_p) \ll_{\text{lexico}} (N_1, \dots, N_q). \end{array}$$

Il est immédiat que $\ll_p$ est un ordre (l'ordre lexicographique et la relation $\ll$ étant des ordres).

On voit que, dans le cas où l'ordre sur A est total, les ordres $\ll$ et $\ll_p$ coïncident.

Comme le montre l'exemple suivant, les ordres $\ll$ et $\ll_p$ ne sont pas équivalents (si l'ordre sur A n'est pas total) :

Exemple :

$$M = \{a, a, 1\} \text{ et } N = \{a, 2\} \text{ (avec } 1 < 2)$$

Les multi-ensembles M et N sont incomparables pour $\ll$. Par contre pour $\ll_p$ on a les partitions <-gauchies suivantes :

$$M_1 = \{a, 1\} \text{ et } M_2 = \{a\}$$

$$N_1 = \{a, 2\}$$

et on a donc :

$$M \ll_p N.$$

On peut démontrer la proposition suivante :

Proposition I.1 [JL 81] :

Soient M et N deux multi-ensembles de $\mathcal{M}(A)$; si $M \ll N$ alors $M \ll_p N$.

Démonstration :

Soient $M \ll N$ et $(M_1, \dots, M_p)$ resp. $(N_1, \dots, N_q)$ les partitions $\leftarrow$ -gauchies de M et N . Montrons que :

$$(M_1, \dots, M_p) \ll_{\text{lexico}} (N_1, \dots, N_q)$$

par récurrence sur $|p|+|q|$.

Cas de base : Pour $M = \{\}$ le résultat est immédiat.

Cas général : Distinguons trois cas :

- Si $M_1 = N_1$, on conclut par hypothèse d'induction sur $M-M_1$ et $N-N_1$.
- Si $M_1 \ll N_1$, alors le résultat est vrai.
- Si $M_1 \gg N_1$ ou M_1 et N_1 sont incomparables, alors il existe nécessairement x dans M_1 tel que $\forall y \in N_1, y \not\leq x$. Cela contredit l'hypothèse $M \ll N$.

On vient donc de définir un ordre sur les multi-ensembles qui est plus général que l'ordre de Manna-Dershowitz. Malheureusement l'ordre $\ll_p$ n'est pas monotone (pour l'extension de l'ordre $\leftarrow$ sur l'ensemble A), tandis que $\ll$ possède cette propriété de monotonie. En effet, en utilisant la définition de Huet-Oppen, on montre facilement la proposition suivante [JL 81] :

Proposition I.2 :

Si l'ordre $\leftarrow$ contient l'ordre $\leftarrow$ sur A , alors l'ordre $\ll$ contient l'ordre $\ll_p$ sur $\mathcal{M}(A)$.

Par contre l'ordre $\ll_p$ ne possède pas cette propriété :

Exemple :

$$M = \{a, a, c\} \text{ et } N = \{a, b, c\}$$

si $b > c$ on a : $M \ll_p N$.

par contre :

si $a > b > c$ on a : $N \ll_p M$.

En fait, on peut montrer qu'il n'existe pas d'ordre qui possède la propriété de monotonie et qui contient strictement l'ordre de Manna-Dershowitz [JL 81].

2.- TERMES

Soit F un ensemble de symboles fonctionnels muni d'une fonctionarité

$$\text{ar} : F \rightarrow \mathbb{N}$$

et soit X un ensemble dénombrable de variables ; par convention : $\text{ar}(x) = 0$ pour tout x dans X .

On note par $T(F, X)$ l'ensemble des termes sur $F \cup X$, encore appelé F -algèbre libre ou magma libre, et par $T(F)$ l'ensemble des termes clos, c'est-à-dire des termes sans variables, encore appelé magma ou algèbre initiale.

Soit $\mathbb{N}_+^*$ l'ensemble des mots sur $\mathbb{N}^*$ muni de l'opération de concaténation notée $\cdot$; ϵ désigne le mot vide.

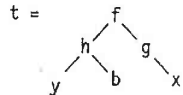
Un terme t de l'ensemble $T(F, X)$ peut être considéré comme un arbre étiqueté défini par la fonction :

$$t : D(t) \subset \mathbb{N}_+^* \rightarrow F \cup X.$$

L'ensemble $D(t)$, appelé ensemble d'occurrences de t , vérifie les propriétés suivantes :

- (i) $\epsilon \in D(t)$;
- (ii) si $u.v \in D(t)$, alors $u \in D(t)$
(propriété préfixe) ;
- (iii) pour tout u de $D(t)$, $u.i \in D(t)$ ssi $i \in [1, \text{ar}(t(u))]$.

Exemple :

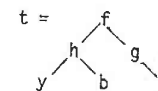


$$\begin{aligned} t(\epsilon) &= f ; & t(11) &= y \\ t(1) &= h ; & t(12) &= b \\ t(2) &= g ; & t(21) &= x \end{aligned}$$

Le sous-arbre de t à l'occurrence $u \in D(t)$ est noté par t/u . Si u n'est pas dans $D(t)$, t/u désigne l'arbre vide noté $\square$. Nous rajoutons ce symbole particulier $\square$ à l'ensemble des symboles fonctionnels F avec la convention : $\text{ar}(\square) = 0$. Lorsque nous définirons une relation d'ordre sur F , elle sera toujours telle que le symbole $\square$ soit plus petit que tout autre symbole fonctionnel.

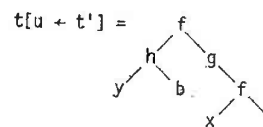
Nous notons par $t[u \leftarrow t']$ pour $u \in D(t)$ le terme t dans lequel on a remplacé le sous-terme de t à l'occurrence u par le terme t' .

Exemple :



$$\text{Alors : } t/1 = \begin{array}{c} h \\ / \quad \backslash \\ y \quad b \end{array}$$

$$\text{Si } t' = \begin{array}{c} f \\ / \quad \backslash \\ x \quad y \end{array} \quad \text{et } u = 21, \text{ alors}$$



Une occurrence u de t est appelée occurrence terminale ou chemin du terme t , si t/u est une feuille de l'arbre, i.e. si $u.i \notin D(t)$ pour tout i dans $\mathbb{N}$.

Le nombre d'occurrences d'un symbole x dans un terme t , noté par $\#(x, t)$, est égal au cardinal de l'ensemble

$$\{u \in D(t) \mid t(u) = x\}.$$

La taille d'un terme t , noté $|t|$, est donné par le cardinal de l'ensemble $\{u \in D(t) \mid t/u \neq \square \text{ et } t(u) \notin X\}$. La taille du terme vide est donc égale à zéro.

Notons par $V(t)$ l'ensemble des variables du terme t , i.e.

$$V(t) = \{x \in X \mid \exists u \in D(t) : t/u = x\}.$$

Pour comparer deux mots u et v de $\mathbb{N}_+^*$, nous utilisons l'ordre préfixe classique :

$u < v$ ssi $\exists w \in \mathbb{N}_+^*$ tel que $w \neq \varepsilon$ et $v = u.w$.

Si $v = u.w$, nous notons par $v/u = w$ le quotient à droite de v par u .

On définit $|u|$ pour u appartenant à $D(t)$ par :

$$|u| = \text{Card}\{p \leq u \mid t/p \neq \square\}$$

Finalement nous définissons le successeur d'une occurrence u préfixe de v par :

$$\text{succ}(u, v) \begin{cases} = u.i & \text{si } v = u.i.w \\ = \infty & \text{si } v = u \end{cases}$$

avec la convention : $t/\infty = \square$.

3.- SYSTEMES DE REECRITURE ET ORDRES DE SIMPLIFICATION

Un système de règles de réécriture est un ensemble d'équations orientées : $R = \{g_i \rightarrow d_i \mid i \in I\}$ telles que pour tout i dans I on ait : $V(d_i) \subseteq V(g_i)$.

Une substitution σ est une application de X dans $T(F, X)$

$\sigma : x \rightarrow \sigma(x)$ telle que $\sigma(x) = x$ sauf pour un nombre fini d'éléments de X .

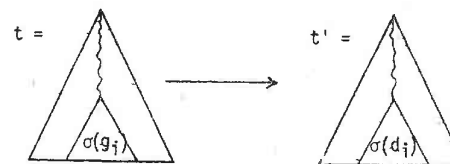
Cette application est étendue par homomorphisme à $T(F, X)$ i.e.

$$\sigma(f(s_1, \dots, s_n)) = f(\sigma(s_1), \dots, \sigma(s_n)).$$

On dit qu'un terme t se réécrit suivant R s'il existe i dans I , u dans $D(t)$ et une substitution σ telles que :

$$t/u = \sigma(g_i).$$

On écrit : $t \xrightarrow[R]{} t'$ avec $t' = t[u + \sigma(d_i)]$ ce qui peut être représenté schématiquement par :



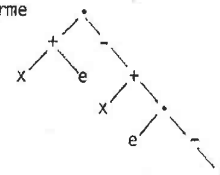
La relation de réduction dans $T(F, X)$, notée $\xrightarrow[R]{*}$, est la fermeture transitive de $\xrightarrow[R]{}.$

Exemple :

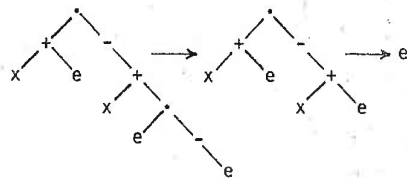
Considérons l'unique règle de réécriture :



et le terme



On obtient successivement :



Un problème important dans les systèmes de réécriture est le problème de la terminaison. Nous savons qu'on ne peut pas décider en général, si la relation $\xrightarrow{*}_R$, définie par au moins deux règles, est bien-fondée sur $T(F,X)$. Nous allons donner maintenant un théorème de N. Dershowitz qui fournit un critère suffisant pour décider si un système de règles de réécriture termine. Ce théorème utilise la notion d'"ordre de simplification".

Définition I.12 :

Un ordre partiel strict $<$ sur $T(F,X)$ est un ordre de simplification s'il possède les deux propriétés suivantes :

- (i) $t < t' \Rightarrow f(\dots, t, \dots) < f(\dots, t', \dots)$ (monotonie)
- (ii) $t < f(\dots, t, \dots)$ (propriété de sous-terme)

Nous pouvons donner maintenant le théorème fondamental de N. Dershowitz :

Théorème I.2 :

Un système de règles de réécritures $R = \{g_i \rightarrow d_i \mid i \in I\}$ comportant un nombre fini de symboles fonctionnels possède la propriété de terminaison finie s'il existe un ordre de simplifica-

tion $<$ vérifiant pour tout i dans I et pour toute substitution σ :

$$\sigma(g_i) > \sigma(d_i).$$

Démonstration :

Voir [DER 79] ; la démonstration est essentiellement basée sur un théorème de Kruskal [KRU 60], [NAS 63].

Mentionnons encore que Dershowitz a généralisé son théorème à des préordres, un préordre étant une relation réflexive et transitive ; le théorème de Dershowitz reste valable si on remplace "ordre de simplification" par "préordre de simplification".

Avec les ordres de simplification on dispose d'un moyen pratique pour prouver la terminaison de systèmes de réécriture. Actuellement on connaît un certain nombre d'ordres de simplification qui sont définis récursivement et pour lesquels il est en général assez facile de montrer quelques propriétés supplémentaires essentielles en pratique, telle la stabilité par instanciation ; dans la suite nous allons rappeler la définition de quelques ordres de simplification importants avant d'introduire et d'étudier une nouvelle famille d'ordres de simplification : les ordres de décomposition.

CHAPITRE II :

ORDRES DE SIMPLIFICATION RÉCURSIFS

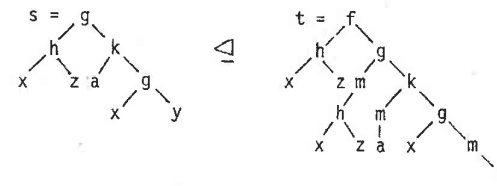
1.- PLONGEMENT

Définition II.1 :

On dit qu'un terme $s = f(\dots, s_i, \dots)$ est plongé dans un terme $t = g(\dots, t_j, \dots)$ et on note $s \triangleleft t$ ssi

- (i) il existe i tel que $s \triangleleft t_i$
- ou (ii) $f = g$ et pour tout $i : s_i \triangleleft t_i$

Exemple



Montrons que le plongement est un ordre de simplification :

Théorème II.1 :

La relation de plongement $\triangleleft$ est un ordre de simplification sur $T(F, X)$ stable par instantiation des variables.

Démonstration :

- a) La réflexivité se montre par induction sur $|t|$: $t \triangleleft t$ car $f = f$ et pour tout $i : t_i \triangleleft t_i$ par induction.
- b) La transitivité de $\triangleleft$ se montre de même par induction sur $|s| + |t| + |u|$.
- c) La propriété de sous-terme est immédiate par (i) et par réflexivité. -

d) La propriété de monotonie est évidente par cas (ii).

e) L'antisymétrie est montré par induction sur $|s|+|t|$:
si $s \trianglelefteq t$ et $t \trianglelefteq s$, alors $s = t$.

Distinguons différents cas :

- Si $f = g$, $s_i \trianglelefteq t_i$ et $t_i \trianglelefteq s_i$ pour tout i , alors on a $s = t$ par induction sur s_i et t_i .
- S'il existe i tel que $s \trianglelefteq t_i$ et il existe j tel que $t \trianglelefteq s_j$, alors on a par la propriété de sous-terme et par transitivité :

$$s \trianglelefteq s_j \text{ et } s_j \trianglelefteq s ;$$

d'où par induction $s = s_j$ ce qui est impossible.

On démontre de même que les deux cas restants conduisent à une impossibilité.

f) Finalement, la stabilité par instanciation des variables se montre facilement par induction sur $|s|+|t|$.

- Si $f = g$ et $s_i \trianglelefteq t_i$, alors $\sigma(s_i) \trianglelefteq \sigma(t_i)$ par induction et donc :

$$\sigma(f(\dots)) \trianglelefteq \sigma(g(\dots)) \text{ par homomorphisme.}$$

- S'il existe i tel que $s \trianglelefteq t_i$, alors par induction :

$$\sigma(s) \trianglelefteq \sigma(t_i)$$

par sous terme : $\sigma(t_i) \trianglelefteq \sigma(t)$

et finalement par transitivité : $\sigma(s) \trianglelefteq \sigma(t)$.

Une propriété essentielle du plongement a été démontrée par Dershowitz [DER 79] :

Proposition II.1 :

Si s et t sont deux termes tels que $s \trianglelefteq t$, alors $s \leq t$ pour tout ordre de simplification.

L'ordre de plongement est donc le plus petit ordre de simplification (au sens de l'inclusion des relations). Cette propriété du plongement sert à construire un exemple pour montrer que la méthode des ordres de simplification n'est pas complète :

Exemple :

Soit la règle de réécriture :

$$\begin{array}{c} a \\ | \\ a \\ | \\ x \end{array} \rightarrow \begin{array}{c} a \\ | \\ b \\ | \\ a \\ | \\ x \end{array}$$

La terminaison de ce système se démontre facilement, mais elle ne peut pas être prouvée à l'aide d'un ordre de simplification, car on a :

$$\begin{array}{c} a \\ | \\ a \\ | \\ x \end{array} \trianglelefteq \begin{array}{c} a \\ | \\ b \\ | \\ a \\ | \\ x \end{array} \quad \text{et donc :} \quad \begin{array}{c} a \\ | \\ a \\ | \\ x \end{array} < \begin{array}{c} a \\ | \\ b \\ | \\ a \\ | \\ x \end{array} \quad \text{pour tout ordre de simplification } <.$$

Différentes implémentations de l'ordre de plongement ont été étudiées dans [CHØ 81].

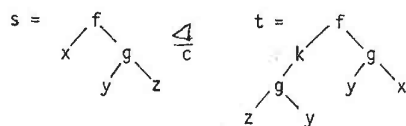
Nous allons donner maintenant deux généralisations de l'ordre de plongement. La première est donnée par Huet-Oppen [HO 80] :

Définition II.2 :

On dit qu'un terme $s = f(s_1, \dots, s_m)$ est plongé commutativement dans un terme $t = g(t_1, \dots, t_n)$ et on note $s \triangleleft_C t$ ssi

- (i) il existe i tel que $s \triangleleft_C t_i$
 - ou (ii) $f = g$ et $m = n$ et il existe une permutation σ de $[1, n]$ tel que
- $$s_i \triangleleft_C t_{\sigma(i)}.$$

Exemple :



Une deuxième extension du plongement est donnée par Dershowitz ; il définit le plongement modulo une relation R [DER 79] :

Définition II.3 :

On dit qu'un terme $s = f(s_1, \dots, s_m)$ est plongé (modulo R) dans un terme $t = g(t_1, \dots, t_n)$ et on note $s \triangleleft_R t$ ssi

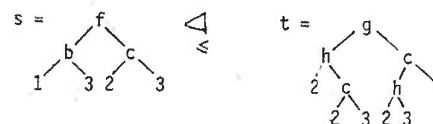
- (i) il existe i tel que $s \triangleleft_R t_i$
 - ou (ii) $f R g$ et pour tout i dans $[1, m]$
- $$s_i \triangleleft_R t_{j_i} \text{ avec } 1 \leq j_1 < j_2 < \dots < j_m \leq n.$$

Si la relation R est l'identité, on retrouve la première

définition du plongement. On voit que, pour que le plongement modulo $R \triangleleft_R$ soit un ordre, il faut que la relation R elle-même soit un ordre.

Exemple :

Si $1 < 2 < 3 \dots$
et $a < b < c \dots$, on a :



Finalement on peut généraliser la proposition II.1 au plongement modulo R . Pour ce faire, introduisons la définition suivante :

Définition II.4 :

Un ordre de simplification $<$ est compatible avec une relation d'ordre R sur l'ensemble F des symboles fonctionnels ssi pour tout f et g dans F avec $ar(f) = ar(g)$ on a :

$$f R g \Rightarrow f(\dots) < g(\dots).$$

Cette définition nous permet d'énoncer la proposition suivante :

Proposition II.2 :

Soit R une relation d'ordre sur F . Si $s = f(\dots s_i \dots)$ et $t = g(\dots t_j \dots)$ sont deux termes tels que $s \triangleleft_R t$, alors $s < t$ pour tout ordre de simplification compatible avec la relation R .

Démonstration :

La démonstration se fait sur $|t|$;

Distinguons deux cas :

- s'il existe i tel que $s \triangleleft_R t_i$, alors par induction :

$s \leq t_i$ et par la propriété de sous-terme : $s \leq t$.

- si $f \in R$ et si pour tout i $s_i \triangleleft_R t_{j_i}$, alors par induction : $s_i \leq t_{j_i}$.

En appliquant n fois la propriété de sous-terme, en utilisant la compatibilité avec R et la transitivité, on obtient :

$$s \leq t$$

2.- ORDRE DE KNUTH-BENDIX

Nous allons définir maintenant l'ordre que Knuth et Bendix ont donné dans leur article présentant un algorithme de complétion [KB 70]. Comme nous le montrerons, il s'agit bien d'un ordre de simplification dont la définition est assez compliquée et demande un certain nombre de restrictions sur l'ensemble des opérateurs.

L'ensemble F des symboles fonctionnels est une suite finie de symboles : $(f_1, \dots, f_N)$ sur laquelle on définit une fonction arité :

$$ar : F \rightarrow \mathbb{N}$$

et une fonction poids : $p : F \rightarrow \mathbb{N}$.

On impose deux conditions à la suite F :

(i) pour tout i dans $[1, N]$: $ar(f_i) = 0 \Rightarrow p(f_i) > 0$

(ii) pour tout i dans $[1, N-1]$: $ar(f_i) = 1 \Rightarrow p(f_i) > 0$

L'ensemble des variables X est supposé distinct de l'ensemble F . Par convention les variables sont supposées d'arité nulle ;

On ne définit pas de poids pour les variables. On note par a_j l'arité de f_j . Nous allons donner d'abord le poids d'un terme clos :

Définition II.5 :

Soit t un terme de $T(F)$. On appelle poids du terme t le nombre $p(t)$ défini par :

$$p(t) = \sum_{i=1}^N p(f_i) \# (f_i, t).$$

On voit que le poids de tout terme clos est strictement positif, les constantes étant de poids strictement positif.

En utilisant la notion de poids, nous définissons un ordre sur les termes de $T(F)$.

Définition II.6 :

Soient s et t deux termes de $T(F)$; on dit que

$$s \triangleleft_{KB} t \text{ ssi}$$

(i) $p(s) < p(t)$

ou (ii) $p(s) = p(t)$ et $s = f_j s_1 \dots s_{a_j}$
 $t = f_k t_1 \dots t_{a_k}$

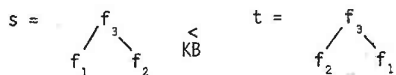
et (iia) $j < k$

ou (iib) $j = k$ et $s_1 = t_1$
 $s_2 = t_2$
 $\vdots$
 $s_{q-1} = t_{q-1}$
 $s_q \triangleleft_{KB} t_q$ pour un $q \in [1; a_k]$.

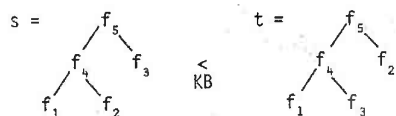
Exemple :

Si $p(f_1) = 1, p(f_2) = 2, p(f_3) = 3 \dots$

on a :



et



Théorème II.2 :

L'ordre $<_{KB}$ de Knuth-Bendix est un ordre de simplification sur $T(F)$.

Démonstration :

a) La relation $<_{KB}$ est un ordre : voir [KB 70].

b) $t <_{KB} f_i(\dots t \dots)$

Si $ar(f_i) \neq 1$ ou si $p(f_i) \neq 0$, on a immédiatement :

$$p(t) < p(f_i(\dots t \dots))$$

et donc $t <_{KB} f_i(\dots t \dots)$

Le seul problème se pose si $f_i = f_N$ avec $p(f_N) = 0$ et $ar(f_N) = 1$; dans ce cas :

$$p(t) = p(f_N(\dots t \dots)).$$

Si le symbole de tête de t est différent de f_N , on conclut par (iia) ; sinon on fait une récurrence sur le nombre de

f_N dans $t = f_N^r t_0$ et $f_N(t) = f_N^{r+1} t_0$; on conclut par (iia) lorsqu'on rencontre le premier symbole f_j différent de f_N avec $j < N$ nécessairement.

$$c) \quad t_1 <_{KB} t_2 \Rightarrow f(\dots t_1 \dots) <_{KB} f(\dots t_2 \dots)$$

Distinguons deux cas :

- si $p(t_1) < p(t_2)$, alors immédiatement :

$$p(f(\dots t_1 \dots)) < p(f(\dots t_2 \dots)) \text{ et donc :}$$

$$f(\dots t_1 \dots) <_{KB} f(\dots t_2 \dots).$$

- si $p(t_1) = p(t_2)$, alors $p(f(\dots t_1 \dots)) = p(f(\dots t_2 \dots))$ et on conclut sur t_1 et t_2 par (iib). ■

De plus Knuth et Bendix démontrent le théorème suivant [KB 70] :

Théorème II.3 :

L'ensemble $T(F)$ des termes clos muni de l'ordre $<_{KB}$ de Knuth-Bendix est bien fondé.

Remarque : Ce théorème peut être obtenu comme corollaire du théorème II.2.

Cet ordre sur les termes clos a été étendu par Knuth et Bendix aux termes avec variables. L'idée de cette extension est la suivante : le poids d'un terme avec variables est égal au poids du terme clos de même forme où chaque variable a été remplacée par la constante de poids minimale.

Si p_0 est le poids de la constante de plus faible poids dans F , on a :

Définition II.7 :

Soit t un terme de $T(F, X)$. On appelle poids du terme t le nombre $p(t)$ défini par :

$$p(t) = p_0 \cdot \sum_{x_i \in X} \#(x_i; t) + \sum_{j=1}^N p_j \#(f_j; t).$$

On peut maintenant donner la définition de l'ordre sur les termes avec variables ; il faudra tenir compte du nombre d'occurrences de chaque variable et dans le cas où $\text{ar}(f_N) = 1$ et $p(f_N) = 0$ des termes de la forme : $f_N^r x_k$; ceci est nécessaire parce que les variables sont incomparables avec les symboles fonctionnels.

Définition II.8 :

Soient s et t deux termes de $T(F, X)$; on dit que

$$s \underset{KB}{<} t \text{ ssi}$$

$$(i) \quad p(s) < p(t) \text{ et } \#(x_i; s) \leq \#(x_i; t) \text{ pour tout } x_i \text{ dans } X$$

$$\text{ou } (ii) \quad p(s) = p(t) \text{ et } \#(x_i; s) = \#(x_i; t) \text{ pour tout } x_i \text{ dans } X$$

et ou bien :

$$s = x_k \text{ et } t = f_N^r x_k \text{ avec } r \geq 1$$

$$\text{ou bien : } s = f_j s_1 \dots s_{a_j} \text{ et } t = f_k t_1 \dots t_{a_k}$$

$$\text{et } (iia) \quad j < k$$

$$\text{ou } (iib) \quad j = k \text{ et } \begin{matrix} s_1 = t_1 \\ \vdots \\ s_{q-1} = t_{q-1} \\ s_q <_{KB} t_q \text{ avec } q \in [1; a_k]. \end{matrix}$$

Exemple :

Si $p(f_1) = 1$; $p(f_2) = 2$ et $p(f_3) = 3$, on a :



par contre :



Finalement le théorème suivant démontré par Knuth et Bendix [KB 70] montre que l'extension de l'ordre aux termes avec variables est cohérente avec la définition initiale et que l'ordre est stable par instantiation des variables :

Théorème II.4 :

Soient s et t deux termes de $T(F, X)$; si $s \underset{KB}{<} t$, alors pour toute substitution σ on a : $\sigma(s) \underset{KB}{<} \sigma(t)$.

3.- ORDRE RECURSIF SUR LES CHEMINS

Avant de donner la définition de l'ordre récursif sur les chemins de Dershowitz, nous introduisons la notion de congruence de

permutation $\equiv^*$ sur $T(F, X)$.

Définition II.9 :

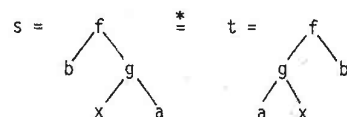
Soient $s = f(s_1, \dots, s_m)$ et $t = g(t_1, \dots, t_n)$ deux termes de $T(F, X)$; on dit que :

$$s \equiv^* t \text{ ssi}$$

$f = g$ et $m = n$ et il existe une permutation σ de $[1, n]$ telle que :

$$s_i \equiv^* t_{\sigma(i)}$$

Exemple :



L'ensemble $T(F, X)/\equiv^*$ est l'ensemble $T(F, X)$ quotienté par la relation $\equiv^*$. Les termes d'une même classe de $T(F, X)/\equiv^*$ sont égaux à une permutation de leurs sous-termes près.

Soit $<$ un ordre partiel sur l'ensemble F des symboles fonctionnels. Donnons d'abord la définition de l'ordre récursif sur les chemins sur les termes clos [DER 79] :

Définition II.10 :

Soient $s = f(s_1, \dots, s_m)$ et $t = g(t_1, \dots, t_n)$ deux termes de $T(F)$; on a :

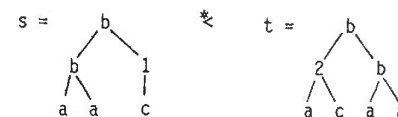
$$s \leq t \text{ ssi}$$

- (i) $f = g$ et $\{s_1, \dots, s_m\} \leq \{t_1, \dots, t_n\}$
- ou (ii) $f < g$ et pour tout i dans $[1, m]$: $s_i \leq t$
- ou (iii) $f \leq g$ et il existe j dans $[1, n]$ tel que $s \leq t_j$ ou $s \equiv^* t_j$.

Remarque : La relation $\leq$ est l'extension de la relation $\leq$ aux multi-ensembles de termes (voir définition I.5 et I.6).

Exemple :

Avec $a < b < c$ et $1 < 2$, on a :



Donnons quelques propriétés de cet ordre récursif sur les chemins (pour les démonstrations voir [DER 79] et [LES 81]).

Théorème II.5 :

La relation $\leq$ est un ordre de simplification sur $T(F)$.

Proposition II.3 :

L'ordre $\leq$ est total sur $T(F)/\equiv^*$ ssi l'ordre $<$ est total sur F .

Proposition II.4 :

Si l'ordre $<$ sur F contient l'ordre $<$, alors $\leq$ sur $T(F)$ contient $\leq$.

Remarquons que cette propriété n'est plus vraie, si on remplace dans (i) de la définition II.10 l'ordre $\leq$ par un ordre

multi-ensemble qui n'est pas monotone pour l'extension de l'ordre sur les symboles, comme l'ordre $\leq_P^*$ de la définition I.11.

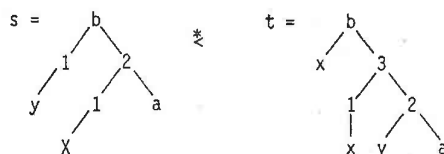
Théorème II.6 [DER 79] :

L'ensemble $(T(F), \leq^*)$ est bien fondé ssi l'ensemble $(F, >)$ est bien fondé.

Pour étendre la définition de $\leq^*$ aux termes avec variables, il suffit d'introduire la convention suivante : toutes les variables sont considérées comme symboles incomparables entre eux et incomparables avec tous les éléments de F pour la relation $<$. Avec cette convention et le cas (iii) de la définition II.10, on définit clairement un ordre sur les termes avec variables.

Exemple :

Avec $a < b < c$ et $1 < 2 < 3$, on a :



Finalement, le théorème suivant montre que l'extension de $\leq^*$ aux termes avec variables est stable par instantiation des variables :

Théorème II.7 :

Soient s et t deux termes de $T(F, X)$; si $s \leq^* t$, alors pour toute substitution σ on a : $\sigma(s) \leq^* \sigma(t)$.

Démonstration :

La preuve se fait par induction sur $|s| + |t|$.

4.- ORDRE RECURSIF LEXICOGRAPHIQUE SUR LES CHEMINS

Kamin et Lévy [KL 80] ont remplacé dans la définition de l'ordre récursif sur les chemins la comparaison de deux multi-ensembles par la comparaison lexicographique de deux listes. Pour garantir la condition de sous-terme de l'ordre ainsi obtenu il faut ajouter une condition supplémentaire dans le cas (i), ce qui nous donne la définition suivante :

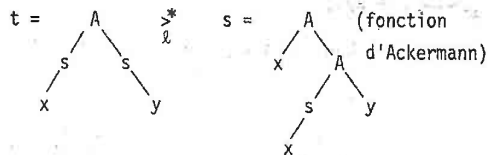
Définition II.11 :

Soient $s = f(s_1, \dots, s_m)$ et $t = g(t_1, \dots, t_n)$ deux termes de $T(F, X)$; on a :

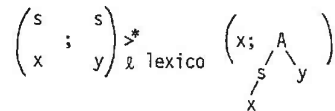
- $s \leq_\ell^* t$ ssi
- (i) $f = g$ et $(s_1, \dots, s_m) \leq_\ell^* (t_1, \dots, t_n)$ lexico et pour tout i dans $[1, m]$: $s_i \leq_\ell^* t_i$
 - ou (ii) $f < g$ et pour tout i dans $[1, m]$: $s_i \leq_\ell^* t_i$
 - ou (iii) $f \not< g$ et il existe j dans $[1, n]$ tel que $s \leq_\ell^* t_j$ ou $s = t_j$.

Remarque : On voit que deux termes s et t d'une même classe de congruence modulo $\leq^*$ ne sont pas équivalents pour l'ordre $\leq_\ell^*$.

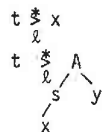
Exemple :



en effet :



et



Remarquons que les termes s et t ci-dessus sont incomparables pour l'ordre $\xrightarrow{\ell}$ de Dershowitz si les symboles s et A sont incomparables.

On peut prouver le résultat :

Théorème II.8 [KL 80] :

La relation $\xrightarrow{\ell}$ est un ordre de simplification sur $T(F,X)$ stable par instantiation des variables.

5.- ORDRE RECURSIF GENERALISE SUR LES CHEMINS

Dans les définitions de l'ordre récursif sur les chemins ou de l'ordre récursif lexicographique sur les chemins on voit que le

choix de l'ordre multi-ensemble ou de l'ordre lexicographique est en quelque sorte arbitraire dans le cas (i).

Pour généraliser ce choix Kamin et Lévy [KL 80] ont introduit une fonctionnelle sur les relations d'ordre c'est-à-dire une application :

$$\triangleright : 2^{T \times T} \longrightarrow 2^{T \times T}$$

On note par $\triangleright$ le résultat de $\triangleright$ appliqué à l'argument $>$. L'ordre multi-ensemble et l'ordre lexicographique sont deux exemples de résultats pour deux fonctionnelles $\triangleright_1$ et $\triangleright_2$.

On peut donner maintenant la définition de l'ordre récursif généralisé sur les chemins :

Définition II.12 :

Soient $s = f(s_1, \dots, s_m)$ et $t = g(t_1, \dots, t_n)$ deux termes de $T(F,X)$; on a :

$$s \xrightarrow[g]{} t \text{ ssi}$$

$$(i) \quad f = g \text{ et } t \triangleright_g^* s \text{ et pour tout } i \text{ dans } [1, m] \\ s_i \xrightarrow[g]{} t$$

$$\text{ou } (ii) \quad f < g \text{ et pour tout } i \text{ dans } [1, m] : s_i \xrightarrow[g]{} t$$

$$\text{ou } (iii) \quad f \not\leq g \text{ et il existe } j \text{ dans } [1, n] \text{ tel que}$$

$$s \xrightarrow[g]{} t_j \text{ ou } s = t_j.$$

Pour que la relation $\xrightarrow[g]{}^*$ soit un ordre de simplification, il faut que la fonctionnelle $\triangleright$ satisfasse certaines conditions :

Théorème II.9 [KL 80] :

La relation $\stackrel{*}{g}$ est un ordre de simplification, si la fonctionnelle $\triangleright$ remplit les conditions suivantes :

- (i) $\triangleright$ préserve l'irréflexivité
- (ii) $\triangleright$ préserve la transitivité
- (iii) si $t > s$, alors $f(\dots t \dots) \triangleright f(\dots s \dots)$
- (iv) $\triangleright$ est continue pour l'inclusion des relations.

Exemple 1 [KL 80] :

Pour orienter le système suivant :

$$A(sx, sy) \rightarrow A(x, A(sx, y))$$

$$A(sx, 0) \rightarrow A(x, so)$$

$$A(0, x) \rightarrow sx$$

- on définit la fonctionnelle $\triangleright$ suivante :

$$A(t_1, t_2) \triangleright_g^* A(t'_1, t'_2) \text{ si } t_1 \stackrel{*}{g} t'_1 \text{ ou } (t_1 = t'_1 \text{ et } t_2 \stackrel{*}{g} t'_2)$$

$$\text{et } st \triangleright_g^* st' \text{ si } t \stackrel{*}{g} t'.$$

- on utilise l'ordre sur les symboles : $A > s$.

Exemple 2 [KL 80] :

Pour orienter la règle d'associativité



on définit :

$$\begin{array}{ccc} \begin{array}{c} \diagup \\ t_1 \end{array} & \triangleright_g^* & \begin{array}{c} \diagup \\ t'_1 \end{array} \\ \begin{array}{c} \diagdown \\ t_2 \end{array} & & \begin{array}{c} \diagdown \\ t'_2 \end{array} \end{array} \quad \begin{array}{l} \text{si } t_1 \stackrel{*}{g} t'_1 \\ \text{ou } \{t_1 = t'_1 \text{ et } \\ t_2 \stackrel{*}{g} t'_2\} \end{array}$$

Remarque : La méthode peut encore se généraliser en partant non plus d'un ordre initial sur F mais d'un ordre sur $T(F, X)$ respectant certaines contraintes. On peut alors sortir du cadre des ordres de simplification étudiés ici.

6.- ORDRE DE PLAISTED

Mentionnons que Plaisted [PLA 78] a défini un ordre sur les termes, appelé ordre sur les chemins de sous-termes, basé sur la notion de "chemin de sous-termes", un chemin de sous-termes d'un terme t étant une liste qui contient t lui-même et un chemin de sous-termes d'un sous-terme t_i de t .

Exemple :

Un chemin de sous-termes de $c = f(x, g(y, c))$ est donné par $[f(x, g(y, c)), g(y, c), c]$.

Nous n'allons pas donner la définition de cet ordre, elle est assez complexe et l'extension de l'ordre aux termes avec variables n'a jamais été donnée explicitement. Disons seulement qu'il s'agit encore d'un ordre de simplification, qui coïncide avec l'ordre $\stackrel{*}{g}$ de Dershowitz, si l'ordre sur les symboles fonctionnels est total.

7.- CONCLUSION

Nous venons de présenter dans ce chapitre un certain nombre d'ordres de simplification définis récursivement. Parmi ces ordres

l'ordre de plongement joue un rôle particulier : c'est le plus petit ordre de simplification. L'ordre de Knuth et Bendix permet de comparer plus de termes, mais sa définition est assez lourde ; en plus il faut imposer un certain nombre de restrictions aux symboles de fonction ce qui en rend l'utilisation délicate. Par contre, la définition de l'ordre récursif sur les chemins de Dershowitz est élégante et simple. L'ordre sur les symboles peut être partiel et l'extension de la définition aux termes avec variables se fait sans problème. Tous ces avantages font que l'ordre récursif sur les chemins se prête bien à comparer deux termes pour un ordre donné sur les symboles. Un autre avantage important de cet ordre est qu'il est monotone pour l'extension de l'ordre sur les symboles de fonction. Ceci est une condition essentielle pour pouvoir utiliser un ordre dans le but d'orienter des règles dans un système de réécriture. En effet, si on est en présence d'une paire de termes incomparables, il faut essayer de compléter l'ordre sur les symboles afin de rendre les termes comparables et de permettre ainsi l'orientation de la règle. Et c'est précisément à ce niveau que se situe le principal désavantage de l'ordre récursif sur les chemins : on ne dispose d'aucune indication pour compléter l'ordre sur les symboles. Ceci vient du fait que l'algorithme ne s'arrête pas, si on tombe sur deux symboles fonctionnels incomparables et que, en présence de deux termes incomparables, il faut parcourir au moins un terme complètement avant que l'algorithme ne s'arrête. Ceci nous a conduit à définir une autre famille d'ordres qui présentent les mêmes avantages que l'ordre récursif sur les chemins, mais fournissent des indications pour accroître l'ordre sur les symboles dans le but de comparer deux termes auparavant incomparables : il s'agit des ordres de décomposition que nous allons introduire dans le chapitre suivant.

CHAPITRE III :

ORDRES DE DÉCOMPOSITION

1.- DEFINITION DE L'ORDRE DE DECOMPOSITION

Nous allons d'abord définir l'ordre de décomposition sur les termes clos ; une extension aux termes avec variables sera donnée au paragraphe 6. La définition de ce nouvel ordre est basée sur la notion de "décomposition" d'un terme t par rapport à un préfixe p d'un chemin u de $D(t)$.

L'ensemble F des symboles fonctionnels est muni d'un ordre partiel noté $<$. Rappelons qu'on est amené à utiliser le symbole particulier noté $\square \notin F$ avec les conventions : $ar(\square) = 0$ et $\square < f$ pour tout f dans F . Le symbole $\square$ servira à marquer l'endroit de la coupure dans le terme lors d'une décomposition. Il sert donc uniquement à conserver des arités fixes pour les symboles fonctionnels et peut être interprété comme étant l'arbre vide. Cela explique les conventions introduites dans le chapitre I, paragraphe 2.

Définition III.1 :

On appelle décomposition élémentaire d'un terme t de $T(F \cup \{\square\})$ à l'occurrence p le long du chemin u , avec $p \leq u$, le quadruplet :

$$d_p^u(t) = \langle t(p); t/succ(p,u); \{t/p.j | j \in [1, ar(t(p))]\} \rangle$$

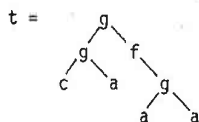
$$\text{et } p.j \neq succ(p,u); t[p \leftarrow \square]$$

$$\text{si } t/p \neq \square$$

$$\text{et } d_p^u(t) = \diamond \text{ si } t/p = \square,$$

où $\diamond$ désigne la décomposition vide.

Exemple :



Nous avons :

$$d_2^{212}(t) = \langle f; \begin{array}{c} g \\ \swarrow \quad \searrow \\ a \quad a \end{array}; \{\}; \begin{array}{c} g \\ \swarrow \quad \searrow \\ c \quad a \end{array} \rangle$$

$$d_1^{11}(t) = \langle g; c; \{a\}; \begin{array}{c} g \\ \swarrow \quad \searrow \\ \square \quad f \\ \quad \quad \swarrow \quad \searrow \\ \quad \quad a \quad a \end{array} \rangle$$

Pour un chemin u donné, nous allons étendre cette définition à un ensemble de préfixes de u . A cet effet nous définissons une application partielle :

$$O : T(F \cup \{\square\}) \times \mathbb{N}_+^* \rightarrow \mathcal{P}(\mathbb{N}_+^*)$$

appelée "application de choix des occurrences préfixes" telle que

$$O(t, u) \subseteq \{p \mid p \leq u\} \text{ pour } u \text{ chemin de } t.$$

Nous avons alors pour un chemin u de t :

$$d_0^u(t) = \{d_p^u \mid p \in O(t, u)\}.$$

$d_0^u(t)$ est donc un ensemble de décompositions élémentaires ; on l'appelle la décomposition de t suivant le chemin u .

De la même manière nous étendons la définition à un ensemble de chemins du terme t par l'application :

$$C : T(F \cup \{\square\}) \rightarrow \mathbb{N}_+^*$$

appelée "application de choix des chemins" et telle que

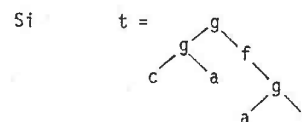
$$C(t) \subseteq \{u \in D(t) \mid \text{ar}(t(u)) = 0\}.$$

Nous obtenons alors :

$$d_0^C(t) = \{d_0^u(t) \mid u \in C(t)\}.$$

L'ensemble $d_0^C(t)$ est un ensemble d'ensembles de décompositions élémentaires, noté $d(t)$ si il n'y a pas d'ambiguïté sur les applications O et C . On l'appelle décomposition du terme t .

Exemple :

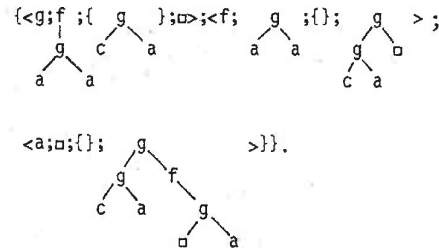


et $C(t) = \{11, 212\}$

$$O(t, 11) = \{11\} \text{ et } O(t, 212) = \{\epsilon, 2, 212\}$$

on a :

$$d(t) = \{ \langle c; \square; \{\}; \begin{array}{c} g \\ \swarrow \quad \searrow \\ \square \quad f \\ \quad \quad \swarrow \quad \searrow \\ \quad \quad a \quad a \end{array} \rangle; \begin{array}{c} g \\ \swarrow \quad \searrow \\ g \quad f \\ \swarrow \quad \searrow \quad \swarrow \quad \searrow \\ a \quad a \quad g \quad a \\ \quad \quad \swarrow \quad \searrow \\ \quad \quad a \quad a \end{array} \rangle \};$$



Nous pouvons maintenant donner la définition de l'ordre de décomposition sur les termes clos. Rappelons que dans un quadruplet de décomposition

$$d_p^u(s) = \langle s(p); s'; \mathcal{Y}'; s'' \rangle$$

- p est le préfixe du chemin le long duquel nous décomposons le terme s ;
- s' est le sous terme s/succ(p,u) ;
- $\mathcal{Y}'$ est le multi-ensemble de tous les autres sous-termes de s commençant en p.i.
- s'' est le terme s[p + a].

Définition III.2 :

Soient s et t deux termes de T(F) et C et O deux applications de choix ; on définit l'ordre de décomposition par

$$s \stackrel{d}{<}_{CO} t \text{ ssi } d_0^C(s) \stackrel{d}{<}_{CO} d_0^C(t)$$

avec $\stackrel{d}{<}_{CO} d_p^u(t)$ pour toute décomposition $d_p^u(t)$ différente de $\diamond$

$$\text{et } d_p^u(s) = \langle s(p); s'; \mathcal{Y}'; s'' \rangle \stackrel{d}{<}_{CO} d_q^v(t) = \langle t(q); t'; \mathcal{Z}'; t'' \rangle$$

ssi on a lexicographiquement :

- (i) $s(p) < t(q)$
- (ii) $d_0^u/succ(p,u)(s') \stackrel{d}{<}_{CO} d_0^v/succ(q,v)(t')$
- (iii) $\mathcal{Y}' \stackrel{d}{<}_{CO} \mathcal{Z}'$
- (iv) $d_0^p(s'') \stackrel{d}{<}_{CO} d_0^q(t'')$

Remarquons que $\stackrel{d}{<}_{CO}$ (respectivement $\stackrel{d}{<}_{CO}$) est obtenu à partir de $\stackrel{d}{<}_{CO}$ (respectivement $\stackrel{d}{<}_{CO}$) en utilisant la définition I.5 ou I.6 ; de même $\stackrel{d}{<}_{CO}$ est obtenu à partir de $\stackrel{d}{<}_{CO}$ en utilisant la même définition ; pour simplifier les notations on écrira dorénavant : $\stackrel{d}{<}_{CO}$, $\stackrel{d}{<}_{CO}$ et $\stackrel{d}{<}_{CO}$.

Remarque : Notons que l'ordre $\stackrel{d}{<}_{CO}$ ne s'applique pas à des multi-ensembles, mais à des ensembles. Un ensemble peut cependant être considéré comme un multi-ensemble particulier et on peut donc comparer deux ensembles à l'aide de l'ordre multi-ensemble ou plus simplement de sa restriction.

La définition de l'ordre de décomposition est une définition récursive basée sur la structure des termes s et t. Montrons que cette définition récursive est bien fondée.

La démonstration se fait par récurrence sur la taille des termes.

Rappelons que

$$|t| = \text{card}\{u \in D(t) \mid t/u \neq \square\}.$$

Le cas de base est obtenu si l'un au moins des deux termes

est de taille nulle, c'est-à-dire si $t = \square$.

Dans le cas général, on compare dans les cas (ii), (iii) et (iv) de la définition des termes plus petits que les termes initiaux :

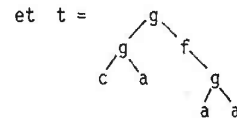
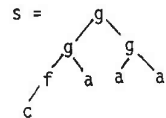
Pour les cas (ii) et (iii) les sous-termes s' et t' et les sous-termes de $\mathcal{S}'$ et $\mathcal{T}'$ sont clairement de taille plus petite que les termes s et t .

Dans le cas (iv) finalement, on a également $|s'| < |s|$ et $|t'| < |t|$; ceci parce qu'on a remplacé un sous-terme de taille au moins égale à un par le terme $\square$ de taille nulle dans le cas où on décompose selon un symbole différent de $\square$. Si on décompose selon $\square$, la décomposition ramène $\langle \rangle$ et l'algorithme s'arrête.

Exemple :

(les choix des chemins et des occurrences préfixes dans cet exemple seront motivés ultérieurement)

Avec $c > a$ et $c > g$ comparons les termes



$$C(s) = \{111\}$$

$$\text{et } C(t) = \{11, 212\}$$

$$O(s, 111) = \{11, 111\}$$

$$\text{et } O(t, 11) = \{11\}$$

$$O(t, 212) = \{\epsilon, 2, 212\}$$

$$d_0^{11}(s) = \{ \langle f; c; \{\} \rangle; \begin{array}{c} g \\ / \quad \backslash \\ g \quad g \\ / \quad \backslash \quad / \quad \backslash \\ \square \quad a \quad a \quad a \end{array} \rangle; \langle c; \square; \{\} \rangle; \begin{array}{c} g \\ / \quad \backslash \\ g \quad f \\ / \quad \backslash \quad / \quad \backslash \\ \square \quad a \quad a \quad a \end{array} \rangle \}$$

$$d_0^{11}(t) = \{ \langle c; \square; \{\} \rangle; \begin{array}{c} g \\ / \quad \backslash \\ g \quad f \\ / \quad \backslash \quad / \quad \backslash \\ \square \quad a \quad a \quad a \end{array} \rangle \}$$

$$d_0^{212}(t) = \{ \langle g; \dots \rangle; \langle f; \begin{array}{c} g \\ / \quad \backslash \\ a \quad a \end{array}; \{\} \rangle; \begin{array}{c} g \\ / \quad \backslash \\ g \quad \square \\ / \quad \backslash \\ c \quad a \end{array} \rangle; \langle a; \dots \rangle \}$$

Alors on a :

$$d_0^{212}(t) \stackrel{d}{<}_{CO} d_0^{11}(s)$$

$$\text{car : } c > a \text{ et } c > g \text{ et } d_0^1(\begin{array}{c} g \\ / \quad \backslash \\ a \quad a \end{array}) \stackrel{d}{<}_{CO} d_0^E(c) .$$

$$\text{De plus nous avons : } d_0^{11}(t) \stackrel{d}{<}_{CO} d_0^{11}(s) .$$

$$\text{Pour } s'' = \begin{array}{c} g \\ / \quad \backslash \\ g \quad g \\ / \quad \backslash \quad / \quad \backslash \\ f \quad a \quad a \quad a \end{array} \text{ nous avons : } d_0^{11}(s'') = \{ \langle g; \begin{array}{c} g \\ / \quad \backslash \\ a \quad a \end{array}; \{\} \rangle; \langle \square; \dots \rangle \}$$

$$\text{Pour } t'' = \begin{array}{c} g \\ / \quad \backslash \\ g \quad f \\ / \quad \backslash \quad / \quad \backslash \\ \square \quad a \quad a \quad a \end{array} \text{ nous avons : } d_0^{11}(t'') = \{ \langle g; \begin{array}{c} g \\ / \quad \backslash \\ a \quad a \end{array}; \{\} \rangle; \langle \square; \dots \rangle \}$$

$$\text{et nous avons : } d_0^1(\begin{array}{c} g \\ / \quad \backslash \\ \square \quad a \end{array}) \stackrel{d}{<}_{CO} d_0^1(\begin{array}{c} g \\ / \quad \backslash \\ f \quad a \end{array})$$

ce qui termine la comparaison avec le résultat : $t \stackrel{d}{<}_{CO} s$.

Remarques :

1) L'ordre d_{CO} est défini même si les chemins u et v n'appartiennent pas à $C(s)$ et $C(t)$ respectivement et si les préfixes p et q n'appartiennent pas à $O(s,u)$ et $O(t,v)$ respectivement.

2) Comme s et t sont des termes de $T(F)$, on peut facilement montrer par récurrence que :

- a) $\mathcal{Y}'$ et $\mathcal{Z}' \in \mathcal{K}(T(F))$.
- b) $\#(\alpha, s') = 0$ ou 1 et alors $s'/(u/succ(p,u)) = \alpha$.
- c) $\#(\alpha, s'') = 1$ et $s''/p = \alpha$.

Il est donc suffisant de définir le choix C pour les termes de $T(F)$.

Nous allons montrer maintenant que l'ordre de décomposition d_{CO} est un ordre partiel strict sur $T(F)$.

Théorème III.1 :

L'ordre de décomposition d_{CO} est une relation anti-réflexive et transitive sur $T(F)$.

Démonstration :

Nous savons que si $<$ est une relation antiréflexive et transitive, l'extension $<<$ de cette relation aux multi-ensembles est encore antiréflexive et transitive [DM 79].

Or, l'ordre de décomposition d_{CO} est une extension à des multi-ensembles des ordres d_{CO} sur les décompositions. Il est

donc suffisant de montrer l'antiréflexivité et la transitivité de la relation d_{CO} sur les quadruplets de décomposition.

La preuve utilise une induction sur la taille des termes. L'antiréflexivité est immédiate à partir de la définition de d_{CO} .

Pour montrer la transitivité considérons les quadruplets :

$$dq_r = \langle f_r; r'; \mathcal{R}'; r' \rangle$$

$$dq_s = \langle f_s; s'; \mathcal{Y}'; s' \rangle$$

$$\text{et } dq_t = \langle f_t; t'; \mathcal{Z}'; t' \rangle$$

$$\text{avec } dq_r \stackrel{d_{CO}}{<} dq_s \text{ et } dq_s \stackrel{d_{CO}}{<} dq_t.$$

Nous distinguons différents cas :

1) Si $f_r < f_s$ ou $f_s < f_t$, alors $f_r < f_t$ et nous avons immédiatement : $dq_r \stackrel{d_{CO}}{<} dq_t$.

2) Si $f_r = f_s = f_t$ et

$$d_0 \cdots (r') \stackrel{d_{CO}}{<} d_0 \cdots (s')$$

$$\text{ou } d_0 \cdots (s') \stackrel{d_{CO}}{<} d_0 \cdots (t')$$

nous utilisons l'hypothèse d'induction pour conclure.

3) Si $f_r = f_s = f_t$ et $d_0 \cdots (r') = d_0 \cdots (s') = d_0 \cdots (t')$

$$\text{et } \mathcal{R}' \stackrel{d_{CO}}{<} \mathcal{Y}' \text{ ou } \mathcal{Y}' \stackrel{d_{CO}}{<} \mathcal{Z}',$$

nous utilisons l'hypothèse d'induction et le fait que l'extension d'une relation aux multi-ensembles conserve la transitivité (si

l'antiréflexivité est garantie).

4) Finalement si les trois premiers éléments des quadruplets de décomposition sont identiques et si

$$d_0^{\dots}(r'') \stackrel{d}{<}_{C_0} d_0^{\dots}(s'')$$

ou

$$d_0^{\dots}(s'') \stackrel{d}{<}_{C_0} d_0^{\dots}(t''),$$

on conclut par hypothèse d'induction.

2.- LES APPLICATIONS DE CHOIX DANS L'ORDRE DE DECOMPOSITION

Dans la définition III.2 on voit que l'ordre de décomposition dépend à tous les niveaux des applications de choix C et O ; nous avons donc défini une famille d'ordres de décomposition indexée par les applications de choix. Dans ce paragraphe nous définissons tout d'abord deux applications de choix particulières : la première consiste à prendre toutes les occurrences préfixes d'un chemin et tous les chemins possibles d'un terme, la deuxième consiste à ne prendre que les chemins et les occurrences maximales dans un sens qui va être précisé. Puis nous étudions les ordres de décomposition correspondant à ces choix particuliers et nous donnons une condition sur les applications de choix pour réduire la famille d'ordres à un ordre unique.

2.1.- Le choix complet et le choix maximal

Définissons d'abord le choix complet ; ce choix est dit complet, parce qu'il prend en considération tous les chemins d'un terme et pour un chemin donné toutes les occurrences préfixes de ce chemin. Notons par C_* et O_* les applications de choix correspondantes. On a alors :

$$C_*(t) = \{u \in D(t) \mid \text{ar}(t(u)) = 0\}$$

$$\text{et } O_*(t,u) = \{p \mid p \leq u\}.$$

Les ensembles d_{0*}^u et d_{0*}^c seront notés d_*^u et d_*^c et on notera $d_{**}^<$ et $d_{**}^<$ les ordres associés.

Avant d'introduire les applications correspondant au choix maximal nous donnons la définition suivante :

Définition III.3 :

Soient E un ensemble muni d'une relation d'ordre $<$ et $M \in \mathcal{M}(E)$; on dit que $C \subset E$ est un ensemble complet d'éléments maximaux ssi pour tout x dans M , il existe y dans C tel que

$$y = x \text{ ou } y > x.$$

On dit que C est l'ensemble complet minimal ssi pour tout y dans C , $C \setminus \{y\}$ n'est pas complet.

Notons que C est unique en tant qu'ensemble ; toutefois si un élément maximal x a plusieurs occurrences dans M , il pourra se poser le problème de savoir de laquelle de ces diverses occurrences provient l'élément x de C .

Exemple :

Avec $1 < 2 < 3 \dots$

et $a < b < c \dots$

et $M = \{c, a, 2, 2, 3, c, 1, b\}$,

$C = \{c, 3\}$ est l'ensemble complet minimal d'éléments maximaux de M .

Nous pouvons maintenant donner la définition de l'applica-

tion de choix maximal qui ne considère que les chemins les plus "importants" d'un terme et qui sera notée par C_+ .

Définition III.4 :

Soit $t = g(t_1, \dots, t_n)$ un terme de $T(F)$; on définit l'application de choix maximal C_+ par :

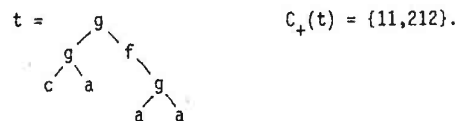
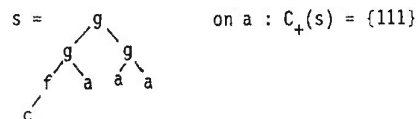
$$C_+(t) = \bigcup_{t_i \in M} C_+(t_i) \text{ avec } M \text{ ensemble complet}$$

minimal de termes maximaux de $(\{t_1, \dots, t_n\}, d_{**}^<)$ et $C_+(a) = \epsilon$ si $ar(a) = 0$.

Remarque : La définition de l'ensemble complet minimal M utilise l'ordre $d_{**}^<$ défini précédemment. On reviendra ultérieurement sur ce problème.

Exemple :

Avec $c > a$ et $c > g$ et



Donnons maintenant la définition de l'application correspondante O_+ : cette application choisit les occurrences préfixes

maximales par comparaison des symboles associés pour l'ordre $<$ sur F .

Définition III.5 :

L'application de choix maximal O_+ est définie par :

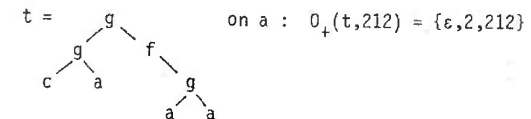
$$O_+(t, u) = \{p \in D(t) \mid p \leq u \text{ et } \forall p' : p' < p \Rightarrow t(p) \not\leq t(p') \\ \forall p'' : p'' > p \Rightarrow t(p) \not\leq t(p'')\}$$

avec t un terme de $T(F)$ et u un chemin de t .

Cette application consiste à choisir sur le chemin u les occurrences p correspondant à des symboles $t(p)$ maximaux pour $<$ sur ce chemin. Si un symbole maximal a plusieurs occurrences sur ce chemin, seule la plus petite est choisie (i.e. celle pour laquelle le symbole est le plus proche de la racine du terme).

Exemple :

Avec $c > a$ et $c > g$ et



Les ensembles $d_{0_+}^u$ et $d_{0_+}^+$ seront notés d_+^u et d_+^+ ; on notera par $d_{++}^<$ et $d_{++}^>$ les ordres associés.

Remarquons que, dans l'exemple à la page 49, nous avons utilisé les applications de choix C_+ et O_+ .

Avant d'étudier en détail les ordres $d_{**}^<$ et $d_{**}^>$, donnons une méthode de calcul pour l'application C_+ . Cette méthode est

basée sur la notion de "gauchissement" d'un terme.

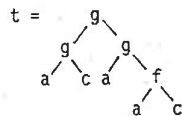
Définition III.6 :

Si $<$ est une relation d'ordre sur $T(F)$ et $t = g(t_1, \dots, t_n)$ un terme de $T(F)$, on dit que t est <-gauchi ssi

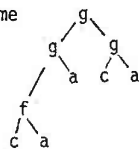
- (i) t_i est <-gauchi pour tout i dans $[1, n]$
- (ii) la liste $(t_1, \dots, t_n)$ est <-gauchie (d'après la définition I.10).

Exemple :

Avec $c > a$ et $c > g$ et



Le terme



est le terme *<-gauchi correspondant.

Si un terme $t = g(t_1, \dots, t_n)$ est <-gauchi, la liste $(t_1, \dots, t_n)$ est une représentation de la partition <-gauchie du multi-ensemble $\{t_1, \dots, t_n\}$. Nous allons utiliser récursivement la classe maximale de cette partition <-gauchie pour définir l'application C_+ sur un terme t .

Définition III.7 :

Soit $t = g(t_1, \dots, t_n)$ un terme de $T(F)$; on définit l'ap-

plication C_+ par :

$$C_+(t) = \bigcup_{i=1}^{i_1} i.C_+(t_i)$$

où l'ensemble $\{t_1, \dots, t_{i_1}\}$ est la classe maximale de la partition $d_{**}^{<}$ -gauchie du multi-ensemble $\{t_1, \dots, t_n\}$.

L'équivalence entre les définitions III.4 et III.7 est immédiate ; il suffit de remarquer que la classe maximale d'une partition gauchie est un ensemble complet minimal d'éléments maximaux.

Dans la suite nous appellerons les éléments de $C_+(t)$ "chemins maximaux" et les éléments de $O_+(t, u)$ "occurrences maximales".

2.2.- La dépendance de l'ordre de décomposition des applications de choix

Nous venons de donner la définition des applications C_* , C_+ , O_* et O_+ qui nous permettent de définir les deux ordres de décomposition $d_{**}^{<}$ et $d_{++}^{<}$, mais aussi les ordres $d_{**}^{<}$ et $d_{++}^{<}$. Nous allons montrer maintenant que les ordres de décomposition ainsi définis sont en fait identiques. De plus, nous démontrerons que la famille d'ordres de décomposition $d_{CO}^{<}$ obtenus en faisant varier les applications C et O se réduit à un unique ordre, si les applications C et O vérifient :

$$(XX) \quad \begin{cases} C(t) \supseteq C_+(t) \text{ où } t \text{ est un terme de } T(F) \\ O(t, u) \supseteq O_+(t, u) \text{ et } u \text{ un chemin de } t. \end{cases}$$

La condition (XX) garantit donc qu'on choisit au moins tous les chemins maximaux et toutes les occurrences maximales pour dé-

composer les termes. Pour montrer que tous les ordres $d_{CO}^<$, où C et O vérifient la condition (XX), sont identiques, il faut montrer que toutes les décompositions le long de chemins non maximaux ou par rapport à des occurrences non maximales engendrent des calculs superflus.

Commençons par démontrer l'inutilité des décompositions suivant des préfixes non maximaux par le lemme suivant :

Lemme III.1 :

Soient s dans $T(F)$, u dans $C_*(s)$ et $p \leq u$. Alors il existe p' dans $O_+(s, u)$ tel que :

$$d_p^u(s) \stackrel{d}{\leq}_{CO} d_{p'}^u(s)$$

pour tout choix vérifiant la condition (XX) ; l'inégalité est stricte si $p' \neq p$.

Démonstration :

Si $p \in O_+(s, u)$, on pose $p' = p$ et on a égalité.

Si $p \notin O_+(s, u)$, alors il existe $p' \in O_+(s, u)$ tel que :

$s(p) \leq s(p')$.

Si $s(p) < s(p')$, le résultat est évident.

Si $s(p) = s(p')$, on a nécessairement $p' < p$. Dans ce cas $s/\text{succ}(p, u)$ est un sous-terme strict de $s/\text{succ}(p', u)$. Le résultat provient alors du lemme suivant qui démontre une propriété de sous-terme.

Lemme III.2 :

Soient s dans $T(F)$, u dans $C_*(s)$ et $k \leq u$. Alors $d_0^{u/k}(s/k) \stackrel{d}{\leq}_{CO} d_0^u(s)$ pour toute application O vérifiant (XX) ; l'inégalité est stricte si $k \neq \epsilon$ c'est-à-dire si s/k

est un sous-terme strict de s .

Démonstration :

La preuve se fait par récurrence sur $|u| - |k|$.

Cas de base : $|u| - |k| = 0$

On distingue différents cas (provenant du fait que l'on peut avoir $u = k.i$ avec $t/u = \square$ et $|u| = |k|$) :

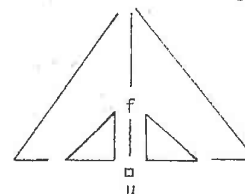
- $u = \epsilon$ et $k = \epsilon$: dans ce cas s se réduit à une constante et on a égalité.

- $s = \frac{f}{\square}$; $u = 1$; si $k = \epsilon$, alors $s/k = \frac{f}{\square}$ et on a encore égalité.

si $k = 1$, alors $s/k = \square$ et l'inégalité est stricte.

- $u \neq \epsilon$ et $u = k$; dans ce cas s/k est un symbole qui occure dans s ; le résultat est vrai s'il existe une décomposition de s qui fait intervenir un symbole au moins aussi grand que $s(u)$: ceci est garanti par la condition (XX).

- $s =$ et $u = k.i$; alors $s/k = f$



Comme dans le cas précédent, la condition (XX) permet de conclure.

Cas général : Soit $p \in O(s/k; u/k)$. Il faut trouver q dans $O(s, u)$ tel que :

$$d_p^{u/k}(s/k) \underset{CO}{d} d_q^u(s).$$

Distinguons deux cas :

1) $k.p \in O(s, u)$; alors $q = k.p$ et la comparaison des deux quadruplets se fait par cas (iv) ; le résultat s'obtient alors par hypothèse d'induction. On a en effet :

$$|k.p| - |k| < |u| - |k|$$

car l'occurrence kp de $s[k.p + \square]$ vérifie $|k.p| \leq |u| - 1$ et l'occurrence du sous-terme est toujours k .

2) $k.p \notin O(s, u)$; alors il existe q dans $O_+(s, u) \subseteq O(s, u)$ (d'après (XX)) tel que $s(q) \geq s(k.p)$. Distinguons à nouveau deux cas :

- a) $s(q) > s(k.p)$ et on conclut par cas (i) de la définition de l'ordre de décomposition.
- b) $s(q) = s(k.p)$; dans ce cas $q < k.p$ et on conclut par cas (ii) en utilisant l'hypothèse de récurrence pour :

$$d_0^{u/k/succ(p, u/k)}(s/k/succ(p, u/k)) \underset{CO}{d} d_0^{u/succ(q, u)}(s/succ(q, u)).$$

en effet : $u/k/succ(p, u/k) = u/succ(kp, u)$ est un sous-chemin de $u/succ(q, u)$ puisque $q < k.p$.
D'autre part : $|u| - [|k.p| + 1] = |u| - |k| - |p| - 1 < |u| - |k|$.
Les hypothèses de récurrence sont donc bien vérifiées.

Si nous notons l'ordre $d_{CO}^<$ par $d_{C*}^<$, nous pouvons démontrer maintenant la proposition suivante :

Proposition III.1 :

Les ordres de décomposition $d_{CO}^<$ et $d_{C*}^<$ sont identiques, si C et O vérifient la condition (XX).

Démonstration :

Si s et t sont deux termes de $T(F)$, il faut montrer que

$$s \underset{CO}{d}^< t \Leftrightarrow s \underset{C*}{d}^< t.$$

La démonstration se fait par récurrence sur $|s| + |t|$.

1) Supposons que $s \underset{CO}{d}^< t$ et montrons que $s \underset{C*}{d}^< t$.

Soit $u \in C(s)$; il existe $v \in C(t)$ tel que

$$d_0^u(s) \underset{CO}{d}^< d_0^v(t).$$

Montrons qu'on a également :

$$d_*^u(s) \underset{C*}{d}^< d_*^v(t).$$

Soit $p \leq u$, il faut trouver $q \leq v$ tel que :

$$d_p^u(s) \underset{C*}{d}^< d_q^v(t).$$

Distinguons deux cas :

a) si $p \in O(s, u)$, il existe $q \in O(t, v)$ tel que

$$d_p^u(s) \underset{CO}{d}^< d_q^v(t) \text{ et par hypothèse d'induction}$$

sur les cas (ii), (iii) ou (iv) de la définition III.2 on a :

$$d_p^u(s) \underset{C_*}{d} d_q^v(t).$$

b) Si $p \notin O(s,u)$, alors d'après le lemme III.1 il existe $p' \in O_+(s,u)$ tel que :

$$d_p^u(s) \underset{CO}{d} d_{p'}^u(s).$$

Or $p' \in O(s,u)$ (d'après la condition (XX)) ; il existe donc $q \in O(t,v)$ tel que

$$d_{p'}^u(s) \underset{CO}{d} d_q^v(t).$$

On a alors par transitivité :

$$d_p^u(s) \underset{CO}{d} d_q^v(t)$$

et on conclut comme dans le cas a) par induction.

2) Supposons que $s \underset{C_*}{d} t$ et montrons que $s \underset{CO}{d} t$.

Soit $u \in C(s)$; il existe $v \in C(t)$ tel que :

$$d_*^u(s) \underset{C_*}{d} d_*^v(t).$$

Montrons qu'on a également :

$$d_0^u(s) \underset{CO}{d} d_0^v(t).$$

Soit $p \in O(s,u)$; il existe $q \leq v$ tel que

$$d_p^u(s) \underset{C_*}{d} d_q^v(t).$$

a) si $q \in O(t,v)$, alors on conclut par cas (i) de la définition III.2 ou en utilisant l'hypothèse d'induction pour les cas (ii), (iii) ou (iv).

b) si $q \notin O(t,v)$, alors il existe d'après le lemme III.1 $q' \in O_+(t,v)$ tel que :

$$d_q^v(t) \underset{C_*}{d} d_{q'}^v(t).$$

D'après la condition (XX), $q' \in O(t,v)$; par transitivité on a :

$$d_p^u(s) \underset{C_*}{d} d_{q'}^v(t)$$

et on conclut comme dans le cas a).

Pour prouver maintenant l'équivalence entre les ordres $\underset{C_*}{d}$ et $\underset{CO}{d}$, nous montrons l'inutilité des décompositions le long de chemins non maximaux par le lemme suivant :

Lemme III.3 :

Soient s un terme de $T(F)$ et u un chemin de s ; si $u \notin C_+(s)$, alors il existe $v \in C_+(s)$ tel que

$$d_*^u(s) \underset{C_*}{d} d_*^v(s).$$

Démonstration :

La démonstration se fait par induction sur $|s|$.

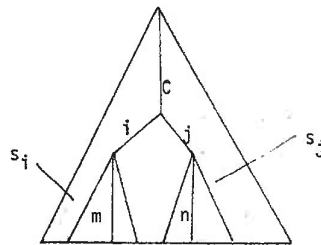
Si $u \in C(t) - C_+(t)$, alors il existe $c, m \in \mathbb{N}_+^*$ et $i \in \mathbb{N}^*$ tel que :

$$-u = c.i.m ;$$

- le sous-terme s/c appartient à la classe maximale des sous-termes à la profondeur c ;
- $s_i = s/c.i$ n'appartient pas à la classe maximale des sous-termes de s/c ;

de plus il existe $j \in \mathbb{N}^*$ tel que

$$s_j = s/c.j \text{ et } s_i \stackrel{d}{\prec}_{C_*} s_j.$$



Il existe donc $n \in C(s_j)$ tel que : $d_*^m(s_i) \stackrel{d}{\prec}_{C_*} d_*^n(s_j)$.

Par induction, on peut supposer que $n \in C_+(s_j)$; il est alors clair que le chemin $c.j.n$ de s appartient à $C_+(s)$.

Posons donc : $v = c.j.n$. Le résultat

$$d_*^{c.i.m}(s) \stackrel{d}{\prec}_{C_*} d_*^{c.j.n}(s)$$

est obtenu par le lemme suivant. ■

Lemme III.3 bis :

Soient s un terme de $T(F)$ et $c, n, m \in \mathbb{N}_+^*$ et $i, j \in \mathbb{N}^*$ tel que $c.i.n$ et $c.j.m$ soient des chemins de s ;

$$\text{si } d_*^m(s/i.m) \stackrel{d}{\prec}_{C_*} d_*^n(s/j.n),$$

$$\text{alors } d_*^{c.i.m}(s) \stackrel{d}{\prec}_{C_*} d_*^{c.j.n}(s).$$

Démonstration :

La preuve se fait par induction sur $|c|+|m|$.

Pour tout préfixe p de $c.i.m$, il faut trouver un préfixe q de $c.j.n$ tel que

$$d_p^{c.i.m}(s) \stackrel{d}{\prec}_{C_*} d_q^{c.j.n}(s).$$

Distinguons trois cas :

1) $p < c$: posons $q = p$; le résultat est vrai alors par cas (ii) de la définition III.2 en utilisant le fait que :

$$\begin{aligned} d_*^{\text{succ}(p, c.i.m)}(s/\text{succ}(p, c.i.m)) &\stackrel{d}{\prec}_{C_*} \\ d^{\text{succ}(p, c.j.n)}(s/\text{succ}(p, c.j.n)) \end{aligned}$$

ce qui est vrai en appliquant l'hypothèse de récurrence à $s/\text{succ}(p, c)$ avec $|c/\text{succ}(p, c)|+|m| < |c|+|m|$.

2) $p = c$; alors $q = c$ et le résultat est vrai par cas (ii) en appliquant l'hypothèse :

$$d_*^m(s_i) \stackrel{d}{\prec}_{C_*} d_*^n(s_j).$$

3) $p > c$: alors il existe $x \in \mathbb{N}_+^*$ tel que $p = c.i.x$ et $y \in \mathbb{N}_+^*$ tel que :

$$d_x^m(s_i) \stackrel{d}{\prec}_{C_*} d_y^n(s_j).$$

Posons alors : $q = c.j.y$ et montrons que :

$$d_{c.i.x}^{c.i.m}(s) \stackrel{d}{\leq}_{C_*} d_{c.j.y}^{c.j.n}(s).$$

Si on compare les deux quadruplets à l'aide des cas (i), (ii) ou (iii), le résultat est immédiat.

Si on utilise le cas (iv), il faut prouver que :

$$d_{*}^{c.i.x}(s[c.i.x + \alpha]) \stackrel{d}{\leq}_{C_*} d_{*}^{c.j.y}(s[c.j.y + \alpha])$$

sachant que :

$$d_{*}^x(s_i[x + \alpha]) \stackrel{d}{\leq}_{C_*} d_{*}^y(s_j[x + \alpha]).$$

On applique l'hypothèse de récurrence avec $|c|+|x| < |c|+|m|$.

En utilisant le lemme III.3 il est maintenant facile de prouver la proposition suivante :

Proposition III.2 :

Les ordres de décomposition $\stackrel{d}{\leq}_{C_*}$ et $\stackrel{d}{\leq}_{**}$ sont identiques, si l'application C vérifie la condition (XX).

Démonstration :

Pour s et t dans $T(F)$ il faut montrer que : $s \stackrel{d}{\leq}_{C_*} t$ ssi $s \stackrel{d}{\leq}_{**} t$. La preuve se fait par récurrence sur $|s|+|t|$.

1) Supposons que $s \stackrel{d}{\leq}_{C_*} t$ et montrons que $s \stackrel{d}{\leq}_{**} t$.

Soit $u \in C(s)$; alors d'après le lemme III.3 il existe $u' \in C_+(s)$ tel que

$$d_{*}^u(s) \stackrel{d}{\leq}_{C_*} d_{*}^{u'}(s).$$

Or $u' \in C(s)$ par la condition (XX) ; il existe donc $v \in C(t)$ tel que

$$d_{*}^{u'}(s) \stackrel{d}{\leq}_{C_*} d_{*}^v(t).$$

Alors par transitivité :

$$d_{*}^u(s) \stackrel{d}{\leq}_{C_*} d_{*}^v(t)$$

avec $v \in C(t)$.

Dans le cas (i), (ii) et (iv) de la définition III.2 l'application C n'intervient pas ; dans le cas (iii) on utilise l'hypothèse d'induction pour obtenir finalement :

$$d_{*}^u(s) \stackrel{d}{\leq}_{**} d_{*}^v(t).$$

2) Supposons que $s \stackrel{d}{\leq}_{**} t$ et montrons que $s \stackrel{d}{\leq}_{C_*} t$.

Soit $u \in C(s)$; par le lemme III.3 il existe $v' \in C_+(t)$ tel que :

$$d_{*}^u(s) \stackrel{d}{\leq}_{**} d_{*}^{v'}(t)$$

et par la condition (XX), $v' \in C(t)$.

On conclut par induction comme dans le cas précédent.

Finalement nous pouvons démontrer le théorème suivant :

Théorème III.2 :

Si les applications de choix C et O vérifient la condition (XX), les ordres de décomposition $\stackrel{d}{\leq}_{C_*}$ et $\stackrel{d}{\leq}_{**}$ sont identiques ; en particulier les ordres $\stackrel{d}{\leq}_{**}^{CO}$ et $\stackrel{d}{\leq}_{++}$ sont identiques.

Démonstration :

On applique successivement les propositions III.1 et III.2.

La définition de $\stackrel{d}{\leq}_{++}$ peut être rendu intrinsèque si on utilise $\stackrel{d}{\leq}_{**}$ pour trier les sous-termes. Pour prouver que ce nou-

vel ordre noté $\overset{0}{<:}_{++}$ est égal à $\overset{0}{<:}_{**}$ on fait le raisonnement suivant :

Si $\overset{0}{<:}_{++} = \overset{0}{<:}_{**}$, alors $\overset{1}{<:}_{++} = \overset{0}{<:}_{++}$ où l'ordre $\overset{1}{<:}_{++}$ est obtenu à partir de $\overset{0}{<:}_{++}$ en triant avec $\overset{0}{<:}_{++}$.

De manière générale :

$$\overset{n}{<:}_{++} = \overset{0}{<:}_{++} \text{ où } \overset{n+1}{<:}_{++} \text{ est obtenu en triant avec } \overset{n}{<:}_{++}.$$

En passant à la limite on obtient :

$$\overset{\infty}{<:}_{++} = \overset{0}{<:}_{++} \text{ à condition que la fonctionnelle soit}$$

continue pour l'inclusion des relations ce qui est une conséquence de la proposition III.4.

Pour faire les preuves dans les paragraphes suivants nous utiliserons surtout l'ordre $\overset{d}{<:}_{**}$. Par contre l'ordre $\overset{d}{<:}_{++}$ permet une implémentation plus efficace de l'ordre de décomposition. Dans la suite nous noterons par $\overset{d}{<:}$ tout ordre de décomposition dont les applications de choix vérifient la condition (XX).

3.- L'ORDRE DE DECOMPOSITION EST UN ORDRE DE SIMPLIFICATION

Pour montrer que l'ordre de décomposition $\overset{d}{<:}$ est un ordre de simplification nous démontrons les deux lemmes suivants :

Lemme III.4 :

Si t est un terme de $T(F)$ on a :

$$t \overset{d}{<:} f(\dots t \dots) \text{ (propriété de sous-terme).}$$

Démonstration :

Soit $u \in C_*(t)$; soit $i \in \mathbb{N}^*$ tel que $f(\dots t \dots)/i = t$. On pose $v = i.u$ et on a

$$d_*^u(t) \overset{d}{<:}_{**} d^{i.u}(f(\dots t \dots))$$

en appliquant le lemme III.2 .

Remarque : Dans la démonstration ci-dessus on a utilisé l'ordre $\overset{d}{<:}_{**}$; ceci sera le cas dans toutes les démonstrations qui suivent à moins qu'on ne le précise autrement.

Lemme III.5 :

Si t_1 et t_2 sont deux termes de $T(F)$, on a la propriété de monotonie suivante :

$$t_1 \overset{d}{<:} t_2 \Rightarrow f(\dots t_1 \dots) \overset{d}{<:} f(\dots t_2 \dots).$$

Démonstration :

On utilise une induction sur $|f(\dots t_1 \dots)| + |f(\dots t_2 \dots)|$.

Soit $u \in C_*(f(\dots t_1 \dots))$. Distinguons deux cas :

1) $u = k.u'$ avec $k \in \mathbb{N}^*$ et $f(\dots t_1 \dots)/k = t_1$.

Montrons que :

$$d_*^u(f(\dots t_1 \dots)) \overset{d}{<:}_{**} d_*^u(f(\dots t_2 \dots)).$$

Pour cela on montre que $\forall p \leq u \quad d_p^u(f(\dots t_1 \dots)) \overset{d}{<:}_{**} d_p^u(f(\dots t_2 \dots))$ par induction sur $|p|$:

- si $p = \varepsilon$, on décide par cas (iii) de la définition III.2 et par l'hypothèse du lemme.
- si $p \neq \varepsilon$, on utilise le cas (iv) en appliquant l'hypothèse d'induction.

2) $u = k.u'$ avec $k \in \mathbb{N}^*$ et $f(\dots t_1 \dots)/k = t_1$.
Alors $u' \in C_*(t_1)$ et il existe $v' \in C_*(t_2)$ tel que :

$$d_*^{u'}(t_1) \stackrel{d}{\leq}_{**} d_*^{v'}(t_2).$$

Posons $v = k.v'$ et montrons que $\forall p \leq u \exists q \leq v$ tel que

$$d_p^{u'}(t_1) \stackrel{d}{\leq}_{**} d_q^{v'}(t_2).$$

Faisons une induction sur $|p|+|q|$ où p et q sont des préfixes de u et de v .

Cas de base : $p = \varepsilon$; alors $q = \varepsilon$ et le résultat

$$d_p^{u'}(f(\dots t_1 \dots)) \stackrel{d}{\leq}_{**} d_q^{v'}(f(\dots t_2 \dots))$$

s'obtient directement par :

$$d_*^{u'}(t_1) \stackrel{d}{\leq}_{**} d_*^{v'}(t_2).$$

Cas général : $p \neq \varepsilon$

Alors il existe q' avec $p = kp'$ et $q = kq'$ tel que :

$$d_p^{u'}(t_1) \stackrel{d}{\leq}_{**} d_q^{v'}(t_2).$$

Si on démontre cette inégalité par les cas (i), (ii) ou (iii), le résultat est immédiat, sinon on applique l'hypothèse d'induction au cas (iv).

Un corollaire de ces deux lemmes est le théorème suivant :

Théorème III.3 :

L'ordre de décomposition $\stackrel{d}{\leq}$ est un ordre de simplification sur $T(F)$.

Pour démontrer le théorème ci-dessus nous avons imposé la condition (XX) aux applications de choix C et O ; cette condition (XX) est donc une condition suffisante pour que l'ordre $\stackrel{d}{\leq}$ soit un ordre de simplification. Malheureusement ce n'est pas une condition nécessaire ; en effet, il y a des ordres de décomposition $\stackrel{d}{\leq}$ où C et O ne vérifient pas la condition (XX) et qui sont des ordres de simplification comme le montrent les exemples suivants :

Exemple 1 :

Soit $F = \{f; a\}$ avec $ar(f) = 1$
 $ar(a) = 0$
et $f > a$.

Les termes de $T(F)$ sont de la forme :

$$a \quad f \quad f \quad \dots \quad f \\ \quad \mid \quad \mid \quad \quad \quad \vdots \\ \quad a \quad f \quad \quad \quad f \\ \quad \quad \mid \quad \quad \quad \mid \\ \quad \quad a \quad \quad \quad a$$

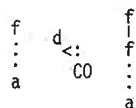
On définit :

$$O(s, u) = \max\{p \leq u \mid s/p \neq \square\}$$

$$C(s) = C_*(s).$$

Remarquons que chaque terme n'a qu'un seul chemin.

L'application O ne vérifie pas la condition (XX) ; cependant $d_{CO}^<$ est un ordre de simplification, car il vérifie clairement la propriété de sous-terme :



On utilise une induction sur le nombre d'occurrences du symbole f et on conclut par le cas (iv) ; le cas de base est :

□ et f .

Exemple 2 :

Soit $F = \{f, a, b\}$ avec $ar(a) = 0$
 $ar(b) = 0$
 $ar(f) = 2$
 et $f > a$
 $f > b$

On définit :

$$O(s, u) = O_+(s, u)$$

L'application C est définie par :

$$C(a) = \{\epsilon\}$$

$$C(b) = \{\epsilon\}$$

$$C\left(\begin{array}{c} f \\ a \quad a \end{array}\right) = \{1\}$$

$$C\left(\begin{array}{c} f \\ b \quad b \end{array}\right) = \{1\}$$

$$C\left(\begin{array}{c} f \\ a \quad b \end{array}\right) = \{2\} !$$

$$C\left(\begin{array}{c} f \\ b \quad a \end{array}\right) = \{1\} !$$

et pour $t = f(t_1, t_2)$ avec $|t| > 3$:

$$C(f(t_1, t_2)) = \{1.C(t_1)\} \cup \{2.C(t_2)\}$$

L'application C ne vérifie donc pas la condition (XX), mais on peut montrer que $d_{CO}^<$ est un ordre de simplification.

a) Propriété de sous-terme :

$$t \quad d_{CO}^< \quad f(t, s)$$

Remarquons que l'unique préfixe par rapport auquel on décompose un terme le long d'un chemin est le préfixe ϵ .

Soit u un chemin du terme t ; on définit

$$\sigma(f; u, s) = \# \{p \leq u \mid s(u) = f\}.$$

On montre que :

$$d_+^u(t) \quad d_{CO}^< \quad d_+^{1 \cdot u}(f(t, s))$$

par induction sur :

$$\sigma(f; u, t) + \sigma(f; 1 \cdot u, f(t, s))$$

Cas de base :

$$a \quad d_{CO}^< \quad \begin{array}{c} f \\ a \quad a \end{array} ; a \quad d_{CO}^< \quad \begin{array}{c} f \\ a \quad b \end{array} \quad \text{etc.}$$

Cas général :

$$d_e^u(t) \quad d_{CO}^< \quad d_e^{1 \cdot u}(f(t, s)) \quad \text{avec}$$

$d_{\varepsilon}^u(t) = \langle f; t / \text{succ}(\varepsilon, u); \dots \rangle$
 et $d_{\varepsilon}^{1 \cdot u}(f(t, s)) = \langle f; t; \dots \rangle$.
 Alors

$$d_{+}^{u / \text{succ}(\varepsilon, u)}(t / \text{succ}(\varepsilon, u)) \stackrel{d}{<}_{CO} d_{+}^u(t)$$

par hypothèse d'induction car :

$$\sigma(f; u / \text{succ}(\varepsilon, u), t / \text{succ}(\varepsilon, u)) = \sigma(f; u, t) - 1$$

$$\text{et } \sigma(f; u, t) = \sigma(f; 1 \cdot u, f(t, s)) - 1.$$

b) Monotonie :

$$t_1 \stackrel{d}{<}_{CO} t_2 \Rightarrow f(t_1, t) \stackrel{d}{<}_{CO} f(t_2, t)$$

Remarquons que $|t_2| \geq 3$, donc que $|f(t_2, t)| > 3$.

Nous distinguons deux cas :

1) Soit u un chemin de $f(t_1, t)$ tel qu'il existe u' avec $u = 2 \cdot u'$; alors $2 \cdot u'$ est un chemin de $f(t_2, t)$ et on a :

$$\begin{aligned} d_{\varepsilon}^{2 \cdot u}(f(t_1, t)) &= \langle f; t; \{t_1\}; \square \rangle \stackrel{d}{<}_{CO} \langle f; t; \{t_2\}; \square \rangle \\ &= d_{\varepsilon}^{2 \cdot u}(f(t_2, t)). \end{aligned}$$

On conclut par cas (iii) et en utilisant l'hypothèse :

$$t_1 \stackrel{d}{<}_{CO} t_2.$$

2) Soit u un chemin de $f(t_1, t)$ tel qu'il existe u' avec $u = 1 \cdot u'$; il existe alors $v' \in C(t_2)$ tel que :

$$d_{+}^{u'}(t_1) \stackrel{d}{<}_{CO} d_{+}^{v'}(t_2).$$

On a alors :

$$d_{+}^{1 \cdot u'}(f(t_1, t)) \stackrel{d}{<}_{CO} d_{+}^{1 \cdot v'}(f(t_2, t)).$$

En effet :

$$\begin{aligned} d_{\varepsilon}^{1 \cdot u'}(f(t_1, t)) &= \langle f; t_1; \{t\}; \square \rangle \stackrel{d}{<}_{CO} \langle f; t_2; \{t\}; \square \rangle \\ &= d_{\varepsilon}^{1 \cdot v'}(f(t_2, t)) \end{aligned}$$

et on conclut par cas (ii). $\square$

Pour conclure on peut donc dire que la condition (XX) n'est pas minimale, puisqu'elle n'est pas nécessaire pour avoir un ordre de simplification.

4.- L'ORDRE DE DECOMPOSITION EST UN ORDRE BIEN FONDE

Nous allons donner deux démonstrations de ce résultat.

Nous utilisons d'abord un théorème de Dershowitz [DER 79] pour prouver la bonne fondation de $\stackrel{d}{<}$. Introduisons la définition suivante :

Définition III.8 [DER 79] :

Un ensemble S est en bonne relation pour une relation R si toute suite infinie $s_1, s_2, \dots$ d'éléments de S contient une paire d'éléments s_i et s_j avec $i < j$ tels que $s_i R s_j$.

Notons qu'un ensemble S est bien fondé pour une relation d'ordre partiel $>$, si et seulement si il est en bonne relation pour la relation $\downarrow$. De plus, si un ensemble est en bonne rela-

tion pour $\leq$, il est bien fondé pour $>$.

Donnons maintenant le théorème de Dershowitz qui permet de prouver la bonne fondation de $d_{<}$.

Théorème III.4 :

Soit $<$: un ordre de simplification sur $T(F)$ qui vérifie la propriété suivante (appelée propriété de remplacement des opérateurs par Dershowitz et qui est en fait une propriété de monotonie) :

$$f < g \Rightarrow f(t_1, \dots, t_m) < g(t_1, \dots, t_m)$$

quels que soient les opérateurs f et g avec $ar(f) = ar(g)$ et les termes $t_1, \dots, t_m$ de $T(F)$. Alors $T(F)$ est en bonne relation pour $\leq$: (donc bien fondé pour $>$), si F est en bonne relation pour $\leq$.

Prouvons la propriété de remplacement des opérateurs pour l'ordre de décomposition $d_{<}$:

Proposition III.3 :

Si f et g sont deux opérateurs de F avec $ar(f) = ar(g)$ et $f < g$, alors $f(\dots) d_{<} g(\dots)$.

Démonstration :

Soit u un chemin de $f(\dots)$; il faut trouver un chemin v de $g(\dots)$ tel que

$$d_{*}^u(f(\dots)) d_{**}^{d_{<}} d_{*}^v(g(\dots)).$$

Preons $v = u$ et faisons une induction sur $|u|$.

Cas de base : $|u| = 1$; on conclut par cas (i) et en utilisant $f < g$.

Cas général : Si $p \leq u$, choisissons $q \leq u$ tel que :

$$d_p^u(f(\dots)) d_{**}^{d_{<}} d_q^u(g(\dots)).$$

Posons $q = p$; si $p = \varepsilon$, le résultat est immédiat, sinon on applique le cas (iv) ; en utilisant l'hypothèse d'induction on montre que :

$$d_{*}^p(f(\dots)[p + \square]) d_{**}^{d_{<}} d_{*}^p(g(\dots)[p + \square]).$$

En utilisant le théorème III.4 et la proposition III.3 on obtient :

Théorème III.5 :

Si F est en bonne relation pour $\leq$, alors $T(F)$ est en bonne relation pour $d_{<}$: (donc bien fondé pour $>$).

Afin de donner un résultat un peu plus général nous utilisons le fait que l'ordre de décomposition $d_{<}$ coïncide avec l'ordre de P. Lescanne, si l'ordre $<$ sur F est total. Rappelons d'abord la définition de l'ordre de P. Lescanne [LES 81].

Soit $<$ un ordre total sur F . Notons que l'ordre récursif sur les chemins $*_{<}$ est total dans $T(F)^{*}$, si l'ordre $<$ sur F est total (proposition II.3). Donnons maintenant la définition de la décomposition (au sens de P. Lescanne) d'un terme $*_{<}$ -gauchi (d'après la définition III.6) :

Définition III.9 :

La décomposition d'un terme s $*_{<}$ -gauchi est l'unique tri-

plet $\langle p; s'; s''(\alpha) \rangle$ tel que :

- p est la plus petite occurrence de l'opérateur maximal de s sur la branche la plus à gauche ;
- $s' = (s'_1, \dots, s'_n)$ est une liste de termes avec $s'_i = s/p.i$;
- $s''(\alpha)$ est le terme $s[p \leftarrow \alpha]$.

Le symbole α joue le même rôle que dans notre définition. Le chemin selon lequel on décompose est le chemin "le plus à gauche", c'est-à-dire le chemin maximal (il est unique parce que l'ordre $*_{<}$ est total) ; de plus on décompose par rapport au préfixe maximal (qui est unique aussi car $<$ sur F est total). L'ordre de P. Lescanne utilise donc implicitement les applications de choix C_+ et O_+ . Remarquons que le terme $s''(\alpha)$ obtenu par la décomposition n'est plus gauchi, mais seulement presque-gauchi dans un sens que nous précisons dans la définition suivante :

Définition III.10 :

Un terme s de $T(F)$ est presque- $*_{<}$ -gauchi ssi

$$s = \alpha$$

$$\text{ou } s = f(s_1, \dots, s_m)$$

et s_i est $*_{<}$ -gauchi pour tout $i \geq 2$.

et s_1 est presque- $*_{<}$ -gauchi

et pour tout i, j avec $1 < i < j \leq m$ $s_j *_{<} s_i$.

Un terme $*_{<}$ -gauchi est évidemment presque- $*_{<}$ -gauchi. Nous pouvons donner maintenant la définition de l'ordre de décomposition de P. Lescanne pour des termes presque- $*_{<}$ -gauchis.

Définition III.11 :

Soient s et t deux termes de $T(F)$ presque $*_{<}$ -gauchis ;

l'ordre de décomposition de P. Lescanne est définie par :

$$s \delta_{<} t \text{ ssi}$$

$$s = \alpha \text{ et } t \neq \alpha$$

ou si on a lexicographiquement :

- (i) $f_s < f_t$
- (ii) $(s'_1, \dots, s'_m) \delta_{\text{lexico}} (t'_1, \dots, t'_n)$
- (iii) $s''(\alpha) \delta_{<} t''(\alpha)$.

Donnons maintenant un résultat important sur l'ordre $\delta_{<}$ [LES 81] :

Théorème III.6 :

L'ordre de décomposition $\delta_{<}$ est un ordre bien fondé sur $T(F)$, si $<$ est un bel-ordre (i.e. un ordre total et bien fondé) sur F . De plus les ordres $\delta_{<}$ et $*_{<}$ sont identiques.

Le lemme suivant montre l'équivalence entre $d_{<}$ et $\delta_{<}$, si l'ordre $<$ sur F est total.

Lemme III.6 :

Si l'ordre $<$ sur F est total, alors l'ordre de décomposition $d_{<}$, l'ordre de décomposition $\delta_{<}$ et l'ordre récursif sur les chemins sont identiques.

Démonstration :

Nous savons déjà que les ordres $*_{<}$ et $\delta_{<}$ sont identiques. Considérons deux termes $*_{<}$ -gauchis $s = f(s_1, \dots, s_m)$ et $t = g(t_1, \dots, t_n)$ et montrons l'équivalence entre $\delta_{<}$ et $d_{<}$. La preuve

se fait par induction sur $|s|+|t|$. Nous utilisons les applications de choix C_+ et O_+ pour faire cette démonstration.

Les sous-termes s_i et t_j sont $d_{<}$ -gauchis par hypothèse d'induction ; les termes s et t sont donc $d_{<}$ -gauchis. En utilisant la définition III.7 on voit que : $C_+(s) = u$ et $C_+(t) = v$ avec u et v les chemins les plus à gauche de s et de t . De plus $O_+(s,u) = \{p\}$ et $O_+(t,v) = \{q\}$ qui sont la plus petite occurrence du symbole maximal de u respectivement de v .

En utilisant ces remarques et l'hypothèse d'induction on voit immédiatement que :

$$d_+^{u/succ(p,u)}(s') d_{++}^{v/succ(q,v)}(t')$$

est équivalent à :

$$s'_i \delta_{<} t'_j.$$

$$\text{Ainsi } [d_+^{p_1}(s') d_{++}^{d_1}(t') \text{ et } \dots] d_{++}^{d_1}(t') \text{ et } \dots]$$

est équivalent à :

$$\langle s'_1, \dots, s'_k \rangle \delta_{<} \langle t'_1, \dots, t'_k \rangle$$

$$\text{et finalement : } d_+^p(s'') d_{++}^{d_1}(s'')$$

est équivalent à :

$$s''(\square) \delta_{<} t''(\square)$$

ce qui termine la démonstration.

Afin de pouvoir donner un résultat analogue à celui du théorème III.6, nous démontrons les résultats suivants :

Lemme III.7 :

Si l'ordre $\prec$ contenant l'ordre $<$ est bien fondé, alors l'ordre $<$ est bien fondé.

Démonstration :

Evident.

Proposition III.4 :

L'ordre $d_{<}$ a la propriété de monotonie, c'est-à-dire :
Si l'ordre $\prec$ sur F contient l'ordre $<$, alors l'ordre de décomposition $d_{\prec}$ sur $T(F)$ contient $d_{<}$.

Démonstration :

Nous allons montrer, s et t étant deux termes de $T(F)$, que $s d_{<} t \Rightarrow s d_{\prec} t$ en utilisant une induction sur $|t|$.

Cas de base : $|t| = 1$: le résultat est immédiat par l'hypothèse : $\prec$ contient $<$.

Cas général : $|t| > 1$

Pour $u \in C(s)$ il faut trouver $v \in C(t)$ tel que :

$$d_*^u(s) d_{**}^{d_1} d_*^v(t).$$

Comme $s d_{<} t$, il existe $v \in C(t)$ tel que pour $p \leq u$ et $q \leq v$ on ait :

$$d_p^u(s) d_{**}^{d_1} d_q^v(t).$$

On choisit les mêmes v et q pour la relation $d_{\prec}$.
Si on compare par le cas (i) de la définition III.2, il faut utiliser l'hypothèse : $\prec$ contient $<$.

Dans les cas (ii), (iii) ou (iv) on utilise l'hypothèse d'induction pour conclure.

Nous pouvons maintenant démontrer le théorème suivant :

Théorème III.7 :

L'ordre de décomposition $d_{<}$ est un ordre bien fondé sur $T(F)$, si l'ordre $<$ est bien fondé sur F .

Démonstration :

Pour pouvoir appliquer le théorème III.6 on plonge l'ordre partiel donné $<$ sur F dans un ordre total et on applique les lemmes III.6 et III.7 et la proposition III.4.

Remarque : Le résultat du théorème III.7 est plus général que celui du théorème III.5 : dans le cas d'un ordre bien fondé une chaîne d'éléments incomparables est permise tandis qu'elle ne l'est pas dans le cas d'un ensemble en bonne relation pour un ordre.

5.- COMPARAISON DE L'ORDRE DE DECOMPOSITION AVEC L'ORDRE RECURSIF SUR LES CHEMINS

Nous allons montrer dans ce paragraphe que l'ordre de décomposition contient strictement l'ordre récursif sur les chemins. Nous allons d'abord démontrer quelques lemmes techniques qui montrent que $d_{<}$ est un point fixe de la fonctionnelle définissant l'ordre récursif sur les chemins.

Lemme III.8 :

Si $d_{*}^u(s_i) d_{**}^{d_{<}} d_{*}^v(g(t_1, \dots, t_n))$ et si $f < g$,

alors $d_{*}^{iu}(f(s_1, \dots, s_i, \dots, s_m)) d_{**}^{d_{<}} d_{*}^v(g(t_1, \dots, t_n))$.

Démonstration :

La preuve se fait par induction sur $|s_i|$.

Cas de base : $|s_i| = 0$ et le résultat est immédiat parce que $f < g$.

Cas général : soit $p \leq i$; il faut trouver q tel que

$$d_p^{i.u}(f(s_1, \dots, s_m)) d_{**}^{d_{<}} d_q^v(g(t_1, \dots, t_n)).$$

Distinguons deux cas :

1) $p = \varepsilon$; alors $q = \varepsilon$ et on décide par $f < g$

2) $p = ip'$; alors il existe q' tel que

$$d_p^u(s_i) d_{**}^{d_{<}} d_q^v(g(t_1, \dots, t_n)).$$

Posons $q = q'$; si on fait la comparaison à l'aide des cas (i), (ii) ou (iii), le résultat est immédiat ; dans le cas (iv) on utilise l'hypothèse d'induction pour montrer l'implication :

$$\text{Si } d_{*}^{p'}(s_i[p' + a]) d_{**}^{d_{<}} d_{*}^{q'}(g(t_1, \dots, t_n)[q' + a]),$$

$$\text{alors } d_{*}^{ip'}(f(\dots)[p + a]) d_{**}^{d_{<}} d_q^v(g(\dots)[q + a]).$$

Lemme III.9 :

$$\text{Si } d^u(s_i) d_{**}^{d_{<}} d^v(t_j),$$

$$\text{alors } d^{iu}(f(s_1, \dots, s_i, \dots, s_m)) d_{**}^{d_{<}} d^{iv}(f(t_1, \dots, t_j, \dots, t_m)).$$

Démonstration :

La démonstration se fait par induction sur $|s_i| + |t_j|$.

Distinguons deux cas :

1) La décomposition se fait en $p = \epsilon$; on choisit $q = \epsilon$ et on décide par le cas (ii) de la définition III.2

2) La décomposition se fait en $p = ip'$; il existe alors q' tel que :

$$d_{ip'}^u(s_i) \stackrel{d}{\leq}_{**} d_q^v(t_j)$$

Posons $q = j.q'$; on a alors :

$$d_{ip'}^{ju}(f(\dots s_i \dots)) \stackrel{d}{\leq}_{**} d_{jq'}^{jv}(g(\dots t_j \dots)).$$

Si on décide par le cas (i), (ii) ou (iii), le résultat est immédiat. Dans le cas (iv) on utilise l'hypothèse d'induction pour montrer que :

$$\begin{aligned} d_{ip'}^u(s_i[p' + \alpha]) &\stackrel{d}{\leq}_{**} d_{jq'}^v(t_j[q' + \alpha]) \\ \Rightarrow d_{ip'}^{ju}(f(\dots)[ip' + \alpha]) &\stackrel{d}{\leq}_{**} d_{jq'}^{jv}(g(\dots)[jq' + \alpha]). \end{aligned}$$

Lemme III.10 :

Si $\{s_1, \dots, s_m\} \stackrel{d}{\leq}_{**} \{t_1, \dots, t_m\}$

alors $d_{*}^{1.u}(f(s_1, s_2, \dots, s_m)) \stackrel{d}{\leq}_{**} d_{*}^{1.u}(f(t_1, t_2, \dots, t_m)).$

Démonstration :

La preuve se fait par induction sur $|s_1|$.

Distinguons de nouveau deux cas :

1) Si $p = \epsilon$, le résultat vient immédiatement de l'hypothèse du lemme.

2) Si $p = 1.p'$, on choisit $q = 1.p'$; on ne peut pas décider par les cas (i), (ii) ou (iii) et on utilise le cas (iv) pour conclure par induction sur :

$$d_{*}^{1.p'}(f(s_1[p' + \alpha], \dots)) \stackrel{d}{\leq}_{**} d_{*}^{1.q'}(f(s_1[q' + \alpha], \dots)).$$

Nous pouvons maintenant démontrer le théorème suivant :

Théorème III.8 :

L'ordre de décomposition $\stackrel{d}{\leq}$ contient l'ordre récursif sur les chemins $\stackrel{*}{\leq}$.

Démonstration :

Soient $s = f(s_1, \dots, s_m)$ et $t = g(t_1, \dots, t_n)$ deux termes de $T(F)$. Montrons que :

$$s \stackrel{*}{\leq} t \Rightarrow s \stackrel{d}{\leq} t \text{ par induction sur } |s| + |t|.$$

Distinguons trois cas suivant les cas (i), (ii) et (iii) de la définition II.10.

1) $f = g$ et $\mathcal{P}' = \{s_1, \dots, s_m\} \stackrel{*}{\leq} \{t_1, \dots, t_m\} = \mathcal{Q}'$

Soit u un chemin de s avec $u = i.u'$ et soit $s_i = s/i$ le sous-terme à l'occurrence i .

Comme $\mathcal{P}' \stackrel{*}{\leq} \mathcal{Q}'$, il existe j tel que $s_i \stackrel{*}{\leq} t_j$.

Nous distinguons deux cas :

a) $s_i \stackrel{*}{\leq} t_j$; on peut supposer que les termes s_i et t_j sont gauchis, donc on a : $s_i = t_j$. Le résultat s'obtient alors en appliquant le lemme III.10.

b) $s_i^* < t_j$: par induction on a : $s_i^{d_<} < t_j$.

Il existe v' tel que :

$$d_{**}^{u'}(s_i) < d_{**}^{v'}(t_j).$$

En appliquant le lemme III.9 on obtient le résultat.

2) $f < g$ et pour tout i dans $[1, m]$: $s_i^* < t$.

On obtient par induction : $s_i^{d_<} < t$

Le résultat est immédiat si on applique le lemme III.8.

3) $f \not< g$ et il existe j tel que $s^* < t_j$.

En utilisant l'hypothèse d'induction on obtient :

$$s^{d_<} < t_j$$

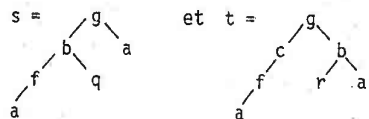
Par la propriété de sous-terme et par transitivité on obtient successivement :

$$t_j^{d_<} < t$$

$$s^{d_<} < t$$

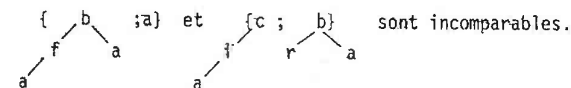
Cependant l'ordre de décomposition $d_<$ et l'ordre $^*<$ ne sont pas équivalents comme le montrent les exemples suivants :

Exemple 1 :



avec $g < b$, $b < c$, $q < r$.

Ces deux termes sont incomparables avec $^*<$, parce que les deux multi-ensembles



Pour $d_<$ nous avons :

$$C_+(s) = \{111, 12\} \text{ et}$$

$$C_+(t) = \{111, 21, 22\}$$

$$\text{et } O_+(s, 111) = \{1, 11, 111\}$$

$$O_+(s, 12) = \{1, 12\}$$

$$O_+(t, 111) = \{1, 11, 111\}$$

$$O_+(t, 21) = \{2, 21\}$$

$$O_+(t, 22) = \{2, 22\}$$

Ainsi on a :

$$d_+^{12}(s) <_{++} d_+^{21}(t)$$

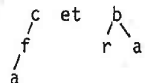
$$\text{et } d_+^{111}(s) <_{++} d_+^{111}(t)$$

ce qui implique : $s^{d_<} < t$.

Essayons de donner une raison intuitive pour ce résultat. Dans le cas de l'ordre $^*<$ un sous-terme du membre gauche doit être majoré par un sous-terme du membre droit sans qu'aucun autre sous-terme du membre droit ne puisse "aider" pour faire la majoration ; par contre dans le cas $d_<$ une aide contrôlée est possible.

Dans l'exemple ci-dessus :

Les deux sous-termes c et b ne peuvent pas dominer



Les sous-termes b et a dans l'ordre $*_{<}$.

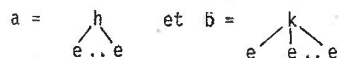


Pour l'ordre de décomposition il n'y a pas de problèmes : un sous-terme est "divisé" en plusieurs chemins qui peuvent être majorés par différents chemins de différents (!) termes.

Exemple 2 :

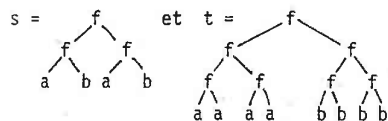
Cet exemple sert à montrer que pour chaque ordre non total $<$ sur F on peut trouver une paire de termes incomparables pour $*_{<}$ et comparables pour $d_{<}$. On suppose qu'il y a un symbole n -aire (avec $n \geq 2$) f et un symbole 0-aire e dans F . Dans la suite, on choisit $n = 2$ sans perte de généralité.

Si h et k sont deux symboles incomparables (il en existe, parce que $<$ n'est pas total), on pose :



On a bien sûr : $a \neq b$.

Soient



Les termes s et t sont incomparables pour l'ordre $*_{<}$, mais on a :

$$s \stackrel{d_{<}}{<} t.$$

6.- EXTENSION DE L'ORDRE DE DECOMPOSITION AUX TERMES NON CLOS

Nous allons donner maintenant une extension de l'ordre de décomposition aux termes avec variables et nous voulons que cette extension soit cohérente avec la définition de l'ordre de décomposition sur les termes clos c'est-à-dire

$$s \stackrel{d_{<}}{<} t \Leftrightarrow \sigma(s) \stackrel{d_{<}}{<} \sigma(t)$$

pour toute substitution σ de X dans $T(F)$.

Nous définissons deux extensions formellement différentes, dont nous démontrons l'équivalence.

Définition III.12 :

La définition de l'application C reste inchangée.

La définition de l'application O est donnée par :

$$O : T(F, X) \times \mathbb{N}_+^* \rightarrow \mathcal{P}(\mathbb{N}_+^*)$$

avec $O(t, u) \subseteq \{p \mid p \leq u\}$ si $t/u \notin X$

et $O(t, u) \subseteq \{p \mid p < u\}$ si $t/u \in X$.

Le principe de comparaison de deux décompositions élémentaires reste identique. Toutefois, on ne pourra comparer deux décompositions suivant les chemins u et v , par exemple

$$d_0^u(s) \stackrel{d_{<}}{<} d_0^v(t)$$

que si il y a compatibilité des variables éventuelles $s(u)$ et $t(v)$, c'est-à-dire :

$$s(u) \notin X \text{ et } d_0^u(s) \stackrel{d_{<}}{<} d_0^v(t)$$

ou bien : $s(u) \in X$ et $s(u) = t(v)$ et $d_0^u(s) \stackrel{d}{<}_{CO} d_0^v(t)$.

La définition III.2 reste inchangée, excepté le cas (ii) qui devient :

(iia) $s'(u/succ(p,u)) \notin X$ et

$$d_0^{u/succ(p,u)}(s') \stackrel{d}{<}_{CO} d_0^{v/succ(q,v)}(t')$$

(iib) $s'(u/succ(p,u)) = t'(v/succ(q,v)) \in X$ et

$$d_0^{u/succ(p,u)}(s') \stackrel{d}{<}_{CO} d_0^{v/succ(q,v)}(t').$$

Définition III.13 :

Les termes de $T(F,X)$ sont considérés comme des termes sur $F \cup X$; les variables sont considérées comme des symboles fonctionnels 0-aires incomparables entre eux et incomparables avec tous les autres symboles de F . La définition de $\stackrel{d}{<}$: reste inchangée.

Proposition III.5 :

Les définitions III.12 et III.13 sont équivalentes.

Démonstration :

La preuve se fait par induction sur $|s|+|t|$.

$$1) s \stackrel{d}{<}_{(13)} t \Rightarrow s \stackrel{d}{<}_{(12)} t$$

Soit $u \in C_*(s)$; il existe $v \in C_*(t)$ tel que :

$$d_*^u(s) \stackrel{d}{<}_{(13)**} d_*^v(t).$$

Il faut distinguer deux cas :

a) Si $s/u \in X$, alors $t/v \in X$, chaque variable étant incomparable avec tout autre symbole. On peut donc comparer s et t par la définition 12. On conclut alors par cas (i) de la définition III.2 ou bien par induction sur les cas (ii), (iii) et (iv).

b) Si $s/u \notin X$, alors $d_p^u(s) \stackrel{d}{<}_{(13)**} d_q^v(t)$ implique que $t/q \notin X$. On termine alors comme dans le cas a).

$$2) s \stackrel{d}{<}_{(12)} t \Rightarrow s \stackrel{d}{<}_{(13)} t$$

De la même manière que sous 1) on déduit de $d_p^u(s) \stackrel{d}{<}_{(13)**} d_q^v(t)$ que $d_p^u(s) \stackrel{d}{<}_{(12)**} d_q^v(t)$ si $s/u \notin X$.

Si $s/u \in X$, alors d'après la définition 12, $s/u = t/v$.

Si $p < u$ on démontre par induction que :

$$d_p^u(s) \stackrel{d}{<}_{(13)**} d_q^v(t)$$

Si $p = u$, on a les quadruplets :

$$\langle X; \emptyset; \{ \}; s[u + \emptyset] \rangle$$

$$\text{et } \langle X; \emptyset; \{ \}; t[u + \emptyset] \rangle.$$

Puisque $d_*^u(s) \stackrel{d}{<}_{(12)**} d_*^v(t)$ avec $s/u = t/v$, on peut terminer par induction.

Démontrons maintenant que ces définitions sont bien cohérentes avec la définition sur les termes clos.

Théorème III.9 :

L'extension de l'ordre de décomposition aux termes avec variables est stable par instantiation.

Démonstration :

Nous utilisons la définition III.12 pour faire la preuve. Nous allons montrer que :

$$s \stackrel{d}{<} t \Rightarrow \sigma(s) \stackrel{d}{<} \sigma(t)$$

par induction sur $|\sigma(s)| + |\sigma(t)|$.

Soit $u \in C_*(\sigma(s))$; il existe u' et u'' dans $\mathbb{N}_+^*$ tel que $u = u' \cdot u''$ et $u' \in C_*(s)$. Comme $s \stackrel{d}{<} t$, il existe $v' \in C_*(t)$ tel que :

$$d_*^{u'}(s) \stackrel{d}{<}_{**} d_*^{v'}(t).$$

Distinguons deux cas :

1) $s/u' \notin X$ alors $u'' = \epsilon$ et $u = u'$.

- si $t/v' \notin X$, on pose $v = v'$ et on a :

$$d_*^u(\sigma(s)) \stackrel{d}{<}_{**} d_*^v(\sigma(t))$$

par cas (i) ou par induction sur les cas (ii), (iii) et (iv) de la définition III.2.

- si $t/v' \in X$, on pose $v = v' \cdot v''$ où v'' est un chemin quelconque de $\sigma(y)$.

Pour majorer la décomposition en $p \leq u \quad d_p^u(\sigma(s))$ on choisit $q \leq v'$ tel que :

$$d_p^u(s) \stackrel{d}{<}_{**} d_q^{v'}(t)$$

et on conclut par cas (i) ou par cas (ii).

2) $s/u' = x \in X$: alors nécessairement $t/v' = x$

- si $\sigma(x) = x$ le résultat est immédiat.
- si $\sigma(x) \neq x$, alors pour tout chemin u'' de $\sigma(x)$ posons $v = v' \cdot u''$; on a alors par induction sur le cas (iv) :

$$d_*^{u' \cdot u''}(\sigma(s)) \stackrel{d}{<}_{**} d_*^{v' \cdot u''}(\sigma(t)).$$

Remarque : Si on utilise la définition 13, la comparaison de deux termes non clos s et t peut échouer après un échec lors de la comparaison des multi-ensembles $d_*^u(s)$ et $d_*^v(t)$; en utilisant la définition 12 on peut éviter parfois cette comparaison en comparant d'abord s/u et t/v si $s/u \in X$. C'est pourquoi nous utilisons toujours dans la pratique la définition 12.

Exemple 1 :

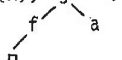
$$s = \begin{array}{c} g \\ / \quad \backslash \\ c \quad y \\ / \quad \backslash \\ x \quad y \end{array} \quad \text{et} \quad t = \begin{array}{c} g \\ / \quad \backslash \\ f \quad a \\ / \quad \backslash \\ c \quad y \\ / \quad \backslash \\ x \quad y \end{array} \quad \text{avec} \quad \begin{array}{l} c > g \\ c > a. \end{array}$$

$$C_+(s) = \{11, 12\} \quad \text{et} \quad C_+(t) = \{111, 112\}.$$

$$d_+^{11}(s) = \{ \langle c; x; \{y\}; \begin{array}{c} g \\ / \quad \backslash \\ \square \quad y \end{array} \rangle \}$$

$$d_+^{12}(s) = \{ \langle c; y; \{x\}; \begin{array}{c} g \\ / \quad \backslash \\ \square \quad y \end{array} \rangle \}$$

$$\text{et} \quad d_+^{111}(t) = \{ \langle f; \dots; \langle c; x; \{y\}; \begin{array}{c} g \\ / \quad \backslash \\ \square \quad y \end{array} \rangle \rangle \}$$

$$d_+^{112}(t) = \{ \langle f; \dots; \langle c; y; \{x\}; \dots \rangle \}$$


Comme : $d_+^1(\text{g}) = \{ \langle g; \alpha; \{y\}; \alpha \rangle \}$



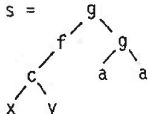
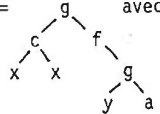
tandis que $d_+^{11}(\text{g}) = \{ \langle f; \dots; \langle g; f; \{a\}; \alpha \rangle \}$



nous avons : $d_+^{11}(s) \stackrel{d}{\leq} d_+^{11}(t)$ et $d_+^{12}(s) \stackrel{d}{\leq} d_+^{112}(t)$;

donc : $s \stackrel{d}{\leq} t$.

Exemple 2 :

$s =$  $\text{et } t =$  avec $c > g$ et $c > a$.

Nous avons :

$$C_+(s) = \{111; 112\} \text{ et } C_+(t) = \{11; 211; 212\}$$

et $d_+^{111}(s) = \{ \langle f; \dots; \langle c; x; \{y\}; \dots \rangle \}$

$$d_+^{11}(t) = \{ \langle c; x; \{x\}; \dots \rangle \}$$

Nous voyons qu'on ne peut pas comparer $d_+^{111}(s)$ et $d_+^{11}(t)$, parce que x et y sont incomparables. Ainsi les termes s et t sont incomparables pour l'ordre de décomposition $\stackrel{d}{\leq}$.

Nous pouvons finalement étendre le théorème III.8 :

Théorème III.10 :

L'extension de l'ordre de décomposition aux termes non clos contient l'extension de l'ordre récursif sur les chemins.

Démonstration :

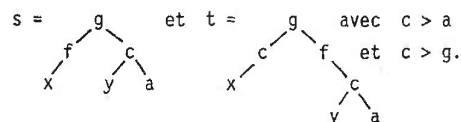
En utilisant la définition III.13 on peut appliquer le théorème III.8, parce que les variables sont considérées comme des constantes incomparables entre elles et avec tous les autres symboles de F .

CONCLUSION

L'ordre de décomposition $d_{<}$ que l'on vient d'étudier est un ordre de simplification bien fondé et monotone pour l'extension de l'ordre $<$ sur les symboles. C'est un ordre qui améliore l'ordre récursif sur les chemins $*_{<}$ de Dershowitz. Malheureusement il n'est pas possible d'orienter des règles comme l'associativité avec $d_{<}$. Cependant, en appliquant une méthode analogue à celle utilisée par Kamin et Lévy [KL 80] pour généraliser l'ordre récursif sur les chemins, on peut espérer obtenir d'autres ordres de décomposition permettant d'orienter la règle d'associativité.

Un avantage majeur de l'ordre de décomposition par rapport à l'ordre récursif sur les chemins est la propriété d'incrémentalité. Nous allons illustrer cette propriété de l'ordre de décomposition sur quelques exemples :

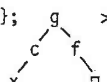
Exemple 1 :



Nous avons :

$$C_+(s) = \{11;21;22\} \text{ et } C_+(t) = \{11;211;212\}$$

$$d_+^{21}(s) = \{<c;y;\{a\}; \text{  >\}$$

$$d_+^{211}(t) = \{<f;...;<c;y;\{a\}; \text{  >\}$$

Nous voyons que :

$$d_+^{21}(s) \stackrel{d_{<}}{<} d_+^{211}(t).$$

De même nous avons :

$$d_+^{22}(s) \stackrel{d_{<}}{<} d_+^{212}(t).$$

Mais il y a échec en comparant :

$$d_+^{11}(s) = \{<g;...;<f;...>\}$$

$$\text{et } d_+^{11}(t) = \{<c;...>\}.$$

Il est cependant facile de voir comment il faut compléter l'ordre sur F pour rendre les deux termes comparables. Il est nécessaire de rendre comparables les symboles f et c .

Ainsi, en imposant $c > f$ on obtient

$$s \stackrel{d_{<}}{<} t.$$

On voit qu'une des différences fondamentales avec l'ordre récursif sur les chemins est le fait que l'ordre de décomposition arrête une comparaison s'il tombe sur deux symboles incomparables.

Exemple 2 :

Cet exemple a été traité par Dershowitz dans [DER 79] pour l'ordre $*_{<}$. Il s'agit d'un système de réécriture de cinq règles qui transforme des formules logiques contenant les symboles $\wedge$, $\vee$ et $\neg$ en des formules équivalentes en forme disjonctive :

$$(i) \quad \neg \neg x \rightarrow x$$

- (ii) $\neg(x \vee y) \rightarrow (\neg x \wedge \neg y)$
- (iii) $\neg(x \wedge y) \rightarrow (\neg x \vee \neg y)$
- (iv) $x \wedge (y \vee z) \rightarrow (x \wedge y) \vee (x \wedge z)$
- (v) $(y \vee z) \wedge x \rightarrow (y \wedge x) \vee (z \wedge x)$

L'orientation de la règle (i) est immédiate par la propriété de sous-terme. Orientons la règle (ii) :

$$s = \begin{array}{c} \neg \\ \swarrow \searrow \\ x \quad y \end{array} \quad d_{<} = \begin{array}{c} \neg \\ \swarrow \searrow \\ x \quad y \end{array} = t$$

$$d_*^{11}(t) = \{ \langle \neg; \begin{array}{c} \neg \\ \swarrow \searrow \\ x \quad y \end{array}; \{\}; \square; \langle \neg; x; \{y\}; \neg \rangle \}$$

$$d_*^{12}(t) = \{ \langle \neg; \begin{array}{c} \neg \\ \swarrow \searrow \\ x \quad y \end{array}; \{\}; \square; \langle \neg; y; \{x\}; \neg \rangle \}$$

$$d_*^{11}(s) = \{ \langle \wedge; \neg; \{ \neg \}; \square; \langle \neg; x; \{ \}; \neg \rangle; \begin{array}{c} \neg \\ \swarrow \searrow \\ \square \quad \neg \\ \quad \swarrow \searrow \\ \quad x \quad y \end{array} \rangle \}$$

$$d_*^{21}(s) = \{ \langle \wedge; \neg; \{ \neg \}; \square; \langle \neg; y; \{ \}; \neg \rangle; \begin{array}{c} \neg \\ \swarrow \searrow \\ \neg \quad \square \\ \swarrow \searrow \\ y \quad x \end{array} \rangle \}$$

Alors on a :

$$d_*^{11}(s) \stackrel{d_{<}}{<} d_*^{11}(t)$$

$$\text{et } d_*^{21}(s) \stackrel{d_{<}}{<} d_*^{12}(t)$$

à condition que : $\neg > \wedge$ ou $\vee > \wedge$.

En échangeant les symboles $\wedge$ et $\vee$ on obtient la condition suivante par la règle (iii) :

$\neg > \vee$ ou $\wedge > \vee$.

Orientons la règle (iv) :

$$s = \begin{array}{c} \vee \\ \swarrow \searrow \\ x \quad y \end{array} \quad d_{<} = \begin{array}{c} \vee \\ \swarrow \searrow \\ x \quad y \end{array} = t$$

$$d_*^{11}(s) = \{ \langle \vee; \begin{array}{c} \vee \\ \swarrow \searrow \\ x \quad y \end{array}; \{ \}; \square; \langle \wedge; x; \{y\}; \neg \rangle; \begin{array}{c} \vee \\ \swarrow \searrow \\ \square \quad \neg \\ \quad \swarrow \searrow \\ \quad y \quad x \end{array} \rangle \}$$

$$d_*^{12}(s) = \{ \langle \vee; \begin{array}{c} \vee \\ \swarrow \searrow \\ x \quad y \end{array}; \{ \}; \square; \langle \wedge; y; \{x\}; \neg \rangle; \dots \rangle \}$$

$$d_*^{21}(s) = \{ \langle \vee; \begin{array}{c} \vee \\ \swarrow \searrow \\ x \quad z \end{array}; \{ \}; \square; \langle \wedge; x; \{z\}; \neg \rangle; \dots \rangle \}$$

$$d_*^{22}(s) = \{ \langle \vee; \begin{array}{c} \vee \\ \swarrow \searrow \\ x \quad z \end{array}; \{ \}; \square; \langle \wedge; z; \{x\}; \neg \rangle; \dots \rangle \}$$

et

$$d_*^1(t) = \{ \langle \wedge; x; \{ \}; \neg; \begin{array}{c} \vee \\ \swarrow \searrow \\ y \quad z \end{array}; \square \rangle \}$$

$$d_*^{21}(t) = \{ \langle \wedge; \begin{array}{c} \vee \\ \swarrow \searrow \\ y \quad z \end{array}; \{x\}; \square; \langle \vee; y; \{z\}; \neg \rangle; \dots \rangle \}$$

$$d_*^{22}(t) = \{ \langle \wedge; \begin{array}{c} \vee \\ \swarrow \searrow \\ y \quad z \end{array}; \{x\}; \square; \langle \vee; z; \{y\}; \neg \rangle; \dots \rangle \}$$

Pour avoir :

$$d_*^{11}(s) \stackrel{d_{<}}{<} d_*^1(t)$$

$$\text{et } d_*^{21}(s) \stackrel{d_{<}}{<} d_*^1(t)$$

il faut que : $\wedge > \vee$.

Avec cette condition on peut terminer avec succès la comparaison de s et de t .

La règle (v) peut être orientée avec la même condition.

En faisant l'intersection de toutes les conditions obtenues on déduit facilement la relation d'ordre suivante :

$$\neg > \wedge > v.$$

Exemple 3 :

Il s'agit d'un autre exemple donné par Dershowitz dans [DER 79].

- (i) $\neg \neg x \rightarrow x$
- (ii) $\neg(x \vee y) \rightarrow ((\neg x \wedge \neg y) \wedge (\neg x \wedge \neg y))$
- (iii) $\neg(x \wedge y) \rightarrow ((\neg x \vee \neg y) \vee (\neg x \vee \neg y))$
- (iv) $(x \wedge x) \rightarrow x$
- (v) $(x \vee x) \rightarrow x$

L'orientation des règles (i), (iv) et (v) est évidente par la propriété de sous-terme.

Orientons la règle (ii) :

$$s = \begin{array}{c} \wedge \\ / \quad \backslash \\ \neg \quad \neg \\ / \quad \backslash \quad / \quad \backslash \\ x \quad y \quad x \quad y \end{array} \stackrel{d}{\prec} \begin{array}{c} \neg \\ | \\ x \vee y \end{array} = t$$

$$d_*^{111}(s) = \{ \langle \wedge; \begin{array}{c} \wedge \\ / \quad \backslash \\ \neg \quad \neg \\ / \quad \backslash \quad / \quad \backslash \\ x \quad y \quad x \quad y \end{array}; \dots; \neg x; \{\}; \dots \rangle \}$$

$$d_*^{121}(s) = \{ \langle \wedge; \begin{array}{c} \wedge \\ / \quad \backslash \\ \neg \quad \neg \\ / \quad \backslash \quad / \quad \backslash \\ x \quad y \quad x \quad y \end{array}; \dots; \neg y; \{\}; \dots \rangle \}$$

$$\text{et} \quad d_*^{11}(t) = \{ \langle \neg; \begin{array}{c} \vee \\ / \quad \backslash \\ x \quad y \end{array}; \{\}; \dots; \neg x; \{y\}; \neg \rangle \}$$

$$d_*^{12}(t) = \{ \langle \neg; \begin{array}{c} \vee \\ / \quad \backslash \\ x \quad y \end{array}; \{\}; \dots; \neg y; \{x\}; \neg \rangle \}.$$

Pour avoir :

$$d_*^{111}(s) \stackrel{d}{\prec} d_*^{11}(t)$$

$$\text{et} \quad d_*^{121}(s) \stackrel{d}{\prec} d_*^{12}(t)$$

il faut que : $v > \wedge$ ou $\neg > \wedge$.

La règle (iii) donne la condition : $\wedge > v$ ou $\neg > v$.

En combinant on obtient la relation : $\neg > \wedge$ et $\neg > v$.

On voit sur ces exemples qu'il est assez facile de trouver la "bonne" relation d'ordre $<$ sur l'ensemble des symboles F permettant d'orienter un système de règles de réécriture en utilisant l'ordre de décomposition $d_{<}$. On peut même espérer pouvoir automatiser cette démarche : en présence d'un certain nombre de règles déduire automatiquement un ordre (minimal si possible) sur les symboles permettant l'orientation des règles dans le sens désiré à l'aide de l'ordre de décomposition. Cependant avant d'écrire un algorithme permettant de résoudre ce problème, il faut donner une implémentation de l'ordre de décomposition afin de pouvoir le comparer avec d'autres ordres de simplification, notamment l'ordre de Dershowitz, du point de vue du temps d'exécution. C'est le développement de ces deux points qui permettra de montrer quelles sont les retombées pratiques de ce travail.

BIBLIOGRAPHIE

Cette bibliographie est volontairement restreinte aux articles pertinents. Une bibliographie complète sur le domaine peut être trouvée dans [HO 80].

- [CHO 81] CHOQUE G. : Algorithmes de calcul de plongement ; rapport CRIN (1981).
- [DER 79] DERSHOWITZ N. : Orderings for Term Rewriting Systems ; Proc. 20 th Symposium on Foundations of Computer Science (1979), p. 123-131.
- [DM 79] DERSHOWITZ N., MANNA Z. : Proving Termination with Multiset Orderings ; Comm. ACM, vol. 22, n° 8, p. 465-476.
- [GOR 67] GORN S. : Explicit Definitions and Linguistic Domains ; "Systems and Computer Science", Ed. Hart J and Takasu S., Univ. of Toronto Press (1967), p. 77-115.
- [HL 78] HUET G., LANKFORD D.S. : On the Uniform Halting Problem for Term Rewriting Systems ; Rapport Laboria 283, IRIA (march 1978).
- [HO 80] HUET G., OPPEN D.C. : Equations and Rewrite Rules : a Survey ; in "Formal Languages perspectives and Open Problems", Ed. Book R., Academic Press (1980).
- [ITT 67] ITTURIAGA R. : Contributions to Mechanical Mathematics ; Ph. D. thesis, Carnegie-Mellon University (1967).

- [JL 81] JOUANNAUD J.P., LESCANNE P. : On Multiset Orderings ; (to appear).
- [KB 70] KNUTH D., BENDIX P. : Simple Word Problems in Universal Algebras ; "Computational Problems in Abstract Algebra", Ed. Leech J., Pergamon Press (1970), p. 263-197.
- [KL 80] KAMIN S., LEVY J.J. : Attempts for Generalizing the Recursive Path Ordering ; (to appear).
- [KRU 60] KRUSKAL J.B. : Well-quasi-ordering, the Tree Theorem and Vazsonyi's Conjecture ; Trans. Amer. Math. Soc. 95 (1960), p. 210-225.
- [LAN 75] LANKFORD D.S. : Canonical Algebraic Simplification : Report ATP-25, Dep. of Math. and Comp. Sciences, University of Texas at Austin (1975).
- [LES 81] LESCANNE P. : A Constructive Proof of the Well-foundedness of the Recursive Path Ordering ; Laboratory for Computer Sciences, MIT (to appear).
- [LR 80] LESCANNE P., REINIG F. : A Well-founded Recursively Defined Ordering on First Order Terms ; rapport interne CRIN 80R005.
- [LS 77] LIPTON R., SNYDER L. : On the Halting of Tree Replacement Systems ; Conference on Theoretical Computer Science, University of Waterloo (1977), p. 43-46.
- [NAS 63] NASH-WILLIAMS C. St. J.A. : On Well-quasi-ordering Finite Trees ; Proc. Cambridge Phil. Soc. 59 (1963), p. 833-835.

- [PLA 78] PLAISTED D. : A Recursively Defined Ordering for Proving Termination of Term Rewriting Systems ; Report 78-943, University of Illinois (september 1978).
- [RJ 81] REINIG F., JOUANNAUD J.P. : Decomposition Orderings : a New Family of Recursive simplification Orderings ; rapport CRIN 81-R-040 (june 1981).



NOM DE L'ETUDIANT : REINIG Fernand

NATURE DE LA THESE : 3ème cycle : Informatique.

VU, APPROUVE

ET PERMIS D'IMPRIMER

NANCY, le 30 OCT, 1981 3246

LE PRESIDENT DE L'UNIVERSITE DE NANCY I, —

Le Vice-Président
de l'Université de Nancy II


F. STRELITZ