UNIVERSITE DE NANCY I

CENTRE DE RECHERCHE EN INFORMATIQUE DE NANCY

CONTRIBUTION A LA RESOLUTION D' EQUATIONS DANS LES ALGEBRES LIBRES ET LES VARIETES EQUATIONNELLES D' ALGEBRES

THESE DE 3e CYCLE EN INFORMATIQUE
PRESENTEE PAR

CLAUDE ET HELENE KIRCHNER

SOUTENUE LE 27 MARS 1982

DEVANT LA COMMISSION D'EXAMEN

PRESIDENT	J.P. JOUANNAUD
EXAMINATEURS	D. BARLET
	B. COURCELLE
	J.P. FINANCE
	G. HUET

UNIVERSITE DE NANCY I

CENTRE DE RECHERCHE EN INFORMATIQUE DE NANCY

CONTRIBUTION A LA RESOLUTION D' EQUATIONS DANS LES ALGEBRES LIBRES ET LES VARIETES EQUATIONNELLES D' ALGEBRES

THESE DE 3e CYCLE EN INFORMATIQUE
PRESENTEE PAR

CLAUDE ET HELENE KIRCHNER

SOUTENUE LE 27 MARS 1982

DEVANT LA COMMISSION D'EXAMEN

PRESIDENT	J.P. JOUANNAUD
EXAMINATEURS	D. BARLET
	B. COURCELLE
	J.P. FINANCE
	G. HUET

REMERCIEMENTS

Ce travail a été réalisé sous la direction de Jean-Pierre Jouannaud dont l'amitié et la compétence ne nous ont jamais fait défaut. Nous le remercions de nous avoir aidés non seulement à développer nos connaissances dans le domaine complexe de la réécriture et des théories équationnelles, mais aussi à formaliser et à étayer nos idées parfois embryonnaires, avec une grande disponibilité et ouverture d'esprit. Nous lui devons également d'avoir pu travailler dans d'excellentes conditions matérielles. Nous lui adressons ici nos sincères remerciements et le remercions aussi de présider ce jury.

Nous sommes vivement reconnaissants à Bruno Courcelle d'avoir critiqué, de facon très constructive et amicale notre travail. Nous le remercions de ses conseils et de sa participation à ce jury.

Gérard Huet nous a aidés non seulement par ses travaux personnels qui sont le fondement d'une partie de notre thèse, mais aussi par ses suggestions et ses critiques judicieuses. Nous le remercions d'avoir accepté de participer à ce jury, malgré ses responsabilités accaparantes à l'INRIA.

Merci également à Jean-Pierre Finance de s'être intéressé à notre travail et d'avoir accepté de lire ces pages.

Daniel Barlet a bien voulu apporter ses compétences de mathématicien à ce jury. Nous l'en remercions sincèrement.

Merci aussi à tous les membres du CRIN qui ont su créer une ambiance de travail efficace et agréable, et plus spécialement à tous ceux qui nous ont suivis dans nos promenades au milieu des arbres.

La réalisation matérielle de cette thèse a été facilitée par l'éditeur de texte développé par Bernard Maréchal, et Danièle Elias a apporté la touche finale à la typographie de ce travail; nous les remercions tous deux de leur disponibilité.

TABLE DES MATIERES

INTRODUCTION 7

PLAN DE LECTURE 11

CHAPITRE 1 :NOTIONS FONDAMENTALES:
THEORIES EQUATIONNELLES ET SYSTEME DE REECRITURE 13

1.1 Théories équationnelles sur des algèbres libres 13
1.1.1 algèbres libres sur un ensemble 13
1.1.2 l'ensemble des arbres étiquetés 15
1.1.2.1 arbres, occurrences, sous-arbres 15
1.1.2.2 structure de M(F,V) 17
1.1.2.3 opérations sur les arbres 18
1.1.2.4 endomorphismes de M(F,V) 19
1.1.3 théorie équationnelle 20
1.1.3.1 équations 20
1.1.3.2 problème de validité. problème des mots 21
1.1.3.3 théorie équationnelle 21
1.1.3.4 théorème de complétude 22
1.1.4 résolution d'équations dans une théorie équationnnelle. 22
1.1.4.1 filtre et préordre de filtrage 22
1.1.4.2 unificateurs dans une théorie équationnelle.... 24
1.1.4.3 ensemble minimal complet de A-unificateurs ... 24
1.1.4.4 unification dans l'ensemble des termes 26
1.1.4.5 ensemble minimal complet de A-filtres 27
1.1.4.6 équations équivalentes 28

1.2 Système de réécriture 30
1.2.1 système de réécriture de termes 31
1.2.2 terminaison d'un système de réécriture de termes 33
1.2.3 confluence d'un système de réécriture 38
1.2.3.1 confluence 38
1.2.3.2 locale confluence et lemme du losange 39
1.2.3.3 paires critiques et théorème de Knuth et Bendix 39
1.2.3.4 système complet ou canonique 41
1.2.3.5 algorithme de complétion 41

1.3 Etude d'extensions de la notion de système de réécriture de termes 47
1.3.1 confluence modulo un ensemble d'équations 48
1.3.1.1 définitions 48
1.3.1.2 le théorème de Huet 49

1.3.2 confluence dans la structure quotient 50
1.3.2.1 généralités 50
1.3.2.2 utilisation d'algorithmes de E-unification ... 52
1.3.2.2.1 la E-compatibilité 53
1.3.2.2.2 paires E-critiques 53
1.3.2.2.3 une condition suffisante de E-confluence 55

CHAPITRE 2 : LES ARBRES SIGNES 57

Introduction

2.1 L'ensemble des arbres signés 58

2.2 Les axiomes et le système de réécriture associé 59

2.3 Etude du système de réécriture E 62

2.4 Complétion du système de réécriture ER associé aux axiomes 63

2.5 Introduction des méta-règles 65
2.5.1 les structures de AS 65
2.5.2 un principe de définition 66
2.5.3 définition de la fonction m 66
2.5.4 propriété de l'opérateur m dans AS 68
2.5.5 définition des méta-règles dans AS 70
2.5.6 ensemble de règles associé à une méta-règle 72

2.6 Le système de réécriture canonique 74

CHAPITRE 3 : RESOLUTION D'EQUATIONS LINEAIRES DANS LES ARBRES SIGNES 83

Introduction

3.1 Equations équivalentes 84

3.2 Simplification d'équations 88
3.2.1 axiomatisation partielle de la notion d'équations équivalentes 89
3.2.2 caractérisation des équations simplifiables 91
3.2.3 équation irréductible obtenue par simplification 95
3.2.4 relation entre les équations irréductibles obtenues par simplification 100

3.3 Equations linéaires 101

3.4 Filtrage dans les arbres signés 104
3.4.1 filtrage et transformation d'équations 104
3.4.2 filtrage dans le cas de termes linéaires 106

TRANSITION 109

CHAPITRE 4: UNE METHODE INCREMENTALE D'UNIFICATION DANS LES THEORIES EQUATIONNELLES 111

Introduction

4.1 La surréduction: un outil pour la résolution d'équations dans les théories équationnelles 112
4.1.1 définition de la surréduction 112
4.1.2 correspondance entre surréduction et réduction 114
4.1.3 relation entre surréduction et unification 118
4.1.4 surréduction basique 119

4.2 La R,E-surréduction: un processus d'unification incrémental 122
4.2.1 définition de la R,E-surréduction 122
4.2.2 la propriété de E-commutation 124
4.2.3 relation entre R,E-réduction et R,E-surréduction 130
4.2.4 unification et R,E-surréduction 135
4.2.5 la R,E-surréduction basique 138

Conclusion 142

CHAPITRE 5: CONSTRUCTION D'UN ALGORITHME DE E-UNIFICATION DANS LES ARBRES SIGNES 145

Introduction

5.1 Un algorithme de E-filtrage dans AS 149

5.2 Un algorithme de E-unification dans AS 151
5.2.1 généralités 151
5.2.2 multiéquations 156
5.2.3 décomposition d'un système de multiéquations 157
5.2.4 fusion d'un système de multiéquations 160
5.2.5 étude des multiéquations telles que $V(e) \cap m(\bar{V}(e)) \neq \emptyset$... 161
5.2.6 la détection des cycles 162
5.2.7 normalisation d'une multiéquation 164
5.2.8 l'algorithme de E-unification 166

Conclusion 171

CHAPITRE 6: UNE DEUXIEME ETUDE DES ARBRES SIGNES 173

Introduction

6.1 Etude de la réductibilité dans l'ensemble quotient 173

6.2 E-commutation dans les arbres signés 174

6.3 La E-confluence 178
6.3.1 E-clôture 179
6.3.2 E-commutation et E-compatibilité 181
6.3.3 terminaison finie 181
6.3.4 E-confluence 181

Conclusion 182

CHAPITRE 7: RESOLUTION D' EQUATIONS QUELCONQUES DANS LES ARBRES SIGNES ... 183

Introduction

7.1 La R,E-surréduction dans les arbres signés 185
7.1.1 quelques exemples 185
7.1.2 la propriété de stricte E-commutation 187

7.2 La méta-R,E-unification 191

7.2.1 définition 191

7.2.2 la méta-E-unification 192
7.2.2.1 l'algorithme de méta-E-unification 192
7.2.2.2 E-unification et méta-E-unification 194

7.2.3 la méta-R,E-surréduction basique dans AS 197
7.2.3.1 définition 198
7.2.3.2 la correction de la méta-R,E-surréduction 199
7.2.3.3 la complétude de la méta-R,E-surréduction 201

7.2.4 description finie d'un ensemble complet de A-solutions d'une équation 204

7.3 Amélioration dans le cas linéaire 207

CHAPITRE 8: APPLICATION A L'INFERENCE DE SEQUENCE DE TERMES 209

Introduction

8.1 Un exemple 211

CONCLUSION 215

ANNEXE : UN OUTIL: UN EDITEUR DE TERMES SOUS FORME ARBORESCENTE 217

BIBLIOGRAPHIE 223

INTRODUCTION

Nous nous intéressons dans cette thèse à différentes méthodes permettant de résoudre des équations dans l'ensemble des termes construits sur un ensemble de symboles de fonctions F et un ensemble de variables V.

Dans la lignée des travaux de R.Boyer et J.Moore [B&M,77], JP.Jouannaud et Y.Kodratoff [J&K,79], les équations qui nous intéressent sont celles qui n'ont pas de solution par la méthode d'unification habituelle, par exemple l'équation f(x,a)=b, où f est un symbole binaire, a et b des constantes et x une variable.

Leur méthode consistait à mêler filtrage (ou unification) et généralisation; celle que nous développons est tout autre: elle consiste à construire l'ensemble des arbres signés, qui est une extension conservative de la F-algèbre libre engendrée par V, et le processus de résolution utilise des transformations algébriques sur les équations, permises par l'introduction d'un nouveau symbole de fonction "-" et d'un ensemble d'axiomes A.

Par exemple, si F est réduit à un unique symbole de fonction binaire f et à un ensemble dénombrable de constantes, on munit les arbres binaires signés, c'est-à-dire les termes construits sur f, -, les constantes et les variables, des axiomes suivants:

$--x = x$; $-f(x,y) = f(-y,-x)$; $f(-y,f(y,x)) = x$; $f(f(x,y),-y) = x$

en vue de pouvoir écrire

$f(x,a) = b \iff f(f(x,a),-a) = f(b,-a) \iff x = f(b,-a)$

et d'obtenir ainsi une solution de l'équation initiale modulo les axiomes A.

Dans le cas de plusieurs symboles de fonctions d'arité quelconque, les axiomes se généralisent aisément et notre premier objectif a été l'étude de la théorie des arbres signés. Nous présentons dans cette thèse les résultats suivants:

* La théorie est décidable. Nous utilisons la méthode de Knuth et Bendix consistant à construire un système de réécriture canonique équivalent aux axiomes.
  Cet algorithme de complétion engendrant un ensemble infini de règles, un nouveau concept, celui de "méta-règle" est introduit, afin de décrire, de facon finie, un ensemble infini de règles, et est utilisé pour faciliter les preuves de confluence.

* Les équations linéaires (dans lesquelles une variable au moins n'apparait qu'une seule fois) ont une solution minimale unique à un isomorphisme près; elle se calcule par transformations d'équations en équations équivalentes. De plus certaines équations non linéaires se réduisent en une équation linéaire équivalente, et sont donc résolubles par la même méthode.

A partir de là, se pose le problème de résoudre les équations non linéaires. Plutôt que de chercher un algorithme de A-unification propre aux arbres signés, nous nous sommes intéressés à la résolution d'équations en général et aux travaux déjà réalisés dans ce domaine.
Nous avons tout d'abord essayé, sans succès, d'étendre la congruence engendrée par les axiomes de la théorie des arbres signés à l'algèbre des arbres infinis, en tentant de suivre les travaux de B.Courcelle [COU,79], de A. Arnold et M. Nivat [A&N,80].

Nous avons ensuite exploré, cette fois avec succès, une seconde voie, en essayant d'appliquer à notre théorie les résultats obtenus par J.M. Hullot sur la surréduction, à la suite de M. Fay et D.S. Lankford. Mais le processus de surréduction d'un terme ne termine pas, en général, dans la théorie des arbres signés. Pour contourner cette difficulté, nous avons alors généralisé la notion de surréduction et cela nous a amenés à construire un algorithme d'unification pour un sous-ensemble des axiomes considérés.

La seconde partie de la thèse a donc deux pôles d'intérêt général:

* Une généralisation de l'algorithme de Martelli et Montanari, proposée par ses auteurs comme un algorithme d'unification dans l'ensemble des termes sans axiomes, à notre théorie équationnelle. Une extension de leur méthode basée sur la décomposition d'une équation en un système d'équations équivalentes, permet de générer des ensembles minimaux complets d'unificateurs. Nous verrons que la méthode que nous proposons est généralisable, et, si elle ne fournit pas toujours un ensemble complet d'unificateurs, elle permet dans tous les cas de simplifier l'équation initiale.

* Une méthode générale d'unification incrémentale: utilisant la connaissance d'un algorithme de E-unification, nous donnons une méthode pour construire un algorithme de A-unification, A étant obtenu en rajoutant aux axiomes de E un ensemble d'axiomes R orientés en règles de réécriture. La méthode utilisée est la R,E-surréduction, notion dejà proposée par D.S.Lankford et A.M.Ballantyne pour des théories associatives et commutatives et qui constitue une généralisation des travaux de J.M.Hullot sur la surréduction. L'algorithme de résolution d'une équation (t=t') consiste à construire l'arbre de toutes les suites de R,E-surréductions possibles faites en parallèle sur les deux termes t et t'. Nous donnons un moyen de se restreindre à certaines R,E-surréductions, dites basiques, permettant d'obtenir la terminaison finie du processus dans certains cas.

La théorie des arbres signés devient alors un exemple non trivial d'application de cette méthode et en illustre la progression.

* C'est ce que nous montrons dans une deuxième étude de la théorie des arbres signés où nous particularisons un sous-ensemble E des axiomes.

Dans le cas binaire, E est constitué des axiomes

$$--x = x \quad \text{et} \quad -f(x,y) = f(-y,-x)$$

et R des règles

$$f(-y,f(y,x)) \longrightarrow x \quad \text{et} \quad f(f(x,y),-y) \longrightarrow x.$$

* Nous montrons alors que $=_E$ est décidable et construisons des algorithmes de E-filtrage et de E-unification complets, ce dernier étant un exemple de la généralisation de l'algorithme de Martelli et Montanari. En général, une équation a une infinité de E-solutions minimales.

Lorsqu'il existe de tels ensembles infinis de E-solutions, l'algorithme de A-unification par R,E-surréduction ne termine pas. Pour lever cette difficulté, dans l'ensemble des arbres signés, nous schématisons l'ensemble infini de E-solutions à l'aide de variables distinguées appelées méta-variables et en déduisons une schématisation d'un arbre de R,E-surréductions ayant une infinité de branches; nous obtenons ainsi un algorithme fini de A-unification dans les arbres signés, permettant de générer un ensemble complet de solutions éventuellement infini.

Nous développons enfin une application des algèbres signées à l'inférence de séquences de termes. Le problème étudié consiste à trouver des relations de récurrence sur une suite de termes, afin d'inférer une description finie de cette suite. Cette méthode a des applications en synthèse et transformation automatique de programme.

Les prolongements de ce travail sont développés dans la conclusion de la thèse.

PLAN DE LECTURE

Les chapitres un et quatre de cette thèse présentent des résultats tout-à-fait généraux.

* Le chapitre un résume les différentes notions utilisées et fixe les notations. Le lecteur familier avec les travaux de G.Huet et J.M.Hullot [H&O,80], [HUL,80] peut en omettre la lecture.

* Le chapitre quatre est consacré à la résolution d'équations par surréduction. Une première partie non originale traite le cas où il existe un système de réécriture canonique, puis une deuxième partie généralise ces résultats à des théories équationnelles définies par un ensemble d'axiomes E et un système de réécriture R, et présente une étude de la R,E-surréduction.

Les chapitres deux et trois d'une part, cinq, six, sept d'autre part, sont davantage tournés vers l'étude de la théorie des arbres signés, dont le chapitre huit donne un exemple d'application.

* Le chapitre deux contient la définition des arbres signés et une première étude de la théorie. Il présente en outre un concept important, celui de "méta-règle", dont nous donnons une formalisation et un exemple d'utilisation dans l'étude de la confluence du système de réécriture infini associé aux axiomes.

* Le chapitre trois est consacré à une axiomatisation partielle de la relation d'équivalence entre équations, à la simplification d'équations en équations équivalentes et à la résolution d'équations linéaires ou linéarisables.

* Le chapitre cinq donne des algorithmes de E-filtrage puis de E-unification; nous dégageons une idée importante qui est la généralisation de l'algorithme de Martelli et Montanari à des théories équationnelles.

* Le chapitre six est une deuxième étude de la théorie des arbres signés en utilisant un système de réécriture E-canonique R.

* Le chapitre sept montre comment appliquer les résultats concernant la R,E-surréduction dans les arbres signés. Nous y formalisons des notions de méta-E-unification et de méta-R,E-surréduction.

* Enfin le chapitre huit est un exemple d'application de la théorie des arbres signés à l'inférence de séquences de termes.

CHAPITRE 1

NOTIONS FONDAMENTALES: THEORIE EQUATIONNELLE ET SYSTEME DE REECRITURE

Nous donnons dans ce chapitre les bases théoriques préliminaires aux thèmes abordés dans cette thèse. Nous y introduisons les termes du premier ordre et donnons divers résultats classiques obtenus en munissant cet ensemble de la structure de F-algèbre.

Nous nous sommes autorisés de nombreux emprunts aux travaux de G. Huet et D.Oppen [H&O,80], M. Bidoit [BID,81], B.Courcelle [COU,81], L.Kott [KOT,80] J.M.Hullot [HUL,80] et renvoyons à ces références le lecteur intéressé par les preuves des résultats cités.

Ce chapitre a également pour but de fixer les notations et la terminologie utilisées dans cette thèse, inspirées de celles de G.Huet et D.Oppen, ainsi que de J.M.Hullot.

1.1- THEORIES EQUATIONNELLES SUR DES ALGEBRES LIBRES

1.1.1- ALGEBRES LIBRES SUR UN ENSEMBLE

Soit F un ensemble dénombrable de symboles de fonctions; à chaque symbole f de F est associé un entier appelé arité de f et noté ar(f). On partitionne F en une réunion de sous-ensembles F_i, où F_i est l'ensemble des symboles de fonctions d' arité i.

CONVENTION: Les symboles de fonctions d'arité 0 sont appelés constantes et notés a,b,c... Ceux d'arité non nulle sont désignés par les lettres f, g, h ...

DEFINITION 1.1: Une F-algèbre est un couple $M=(D_M,F_M)$ où D_M est un ensemble non vide appelé domaine de M, et F_M une famille d'opérations sur D_M indexées par l'ensemble des symboles de fonctions de F. On notera $F_M = \{f_M / f \text{ appartient à } F\}$.

EXEMPLE : Soit $F = \{0,S,+\}$. Choisissons pour domaine l'ensemble N des entiers naturels et pour opérations : 0_N qui est l'entier 0, S_N qui est la fonction successeur, $+_N$ qui est l'addition usuelle.
Le couple $(N,\{0_N,S_N,+_N\})$ est une F-algèbre.

DEFINITION 1.2: Soient $M=(D_M,F_M)$ et $M'=(D_{M'},F_{M'})$ deux F-algèbres. Un homomorphisme h de M dans M' est une application de D_M dans $D_{M'}$ telle que pour tout symbole de fonction f d'arité n dans F et pour tous éléments $d_1, d_2,\ldots, d_n$ dans D_M, on ait : $h(f_M(d_1,d_2,\ldots,d_n)) = f_{M'}(h(d_1),h(d_2),\ldots,h(d_n))$.

Un homomorphisme d'une F-algèbre dans elle même est appelé endomorphisme. Si de plus il est bijectif, il est appelé isomorphisme.

DEFINITION 1.3: Soit A une classe quelconque de F-algèbres et V un ensemble. On appelle F-algèbre A-libre engendrée par V (on dit aussi sur V) toute F-algèbre $M=(D_M,F_M)$ tel que

1) M appartient à A

2) D_M contient l'ensemble V

3) pour toute F-algèbre $N=(D_N,F_N)$ de A et toute application h_0 de V dans D_N, il existe un unique homomorphisme h de M dans N dont la restriction à V soit égale à h_0.

NOTATION : Les éléments de V sont appelés variables et notés dans la suite x, y, z.

Si M_1 et M_2 sont deux F-algèbres A-libres sur V, il est facile de montrer qu'il existe un isomorphisme unique entre M_1 et M_2 qui est l'identité sur V. On peut donc sans ambiguité parler de la F-algèbre A-libre engendrée par V.

Le résultat suivant, cas particulier d'un théorème dû à Birkhoff, nous permet de définir l'ensemble des termes.

THEOREME 1.1 [BIR,35]: Soit A la classe de toutes les F-algèbres pour F fixé et V un ensemble dont les éléments sont appelés variables. On supposera toujours que soit V soit l'ensemble des constantes est non vide. Alors la F-algèbre A-libre (ou plus simplement la F-algebre libre) engendrée par V existe.

DEFINITION 1.4: Dans les conditions du théorème précédent, on appelle ensemble des termes du premier ordre, ou plus simplement termes, les éléments de la F-algèbre A-libre engendrée par V.

EXEMPLE : Si A est l'ensemble des groupes, on obtient la définition du groupe libre engendré par un ensemble V.

NOTATION : Les termes seront désignés dans la suite par u, v, w, t, t',..., g, d,... .

Cette définition des termes étant particulièrement peu explicite et peu parlante à notre intuition, nous allons en donner une représentation dans le paragraphe suivant.

1.1.2- L'ENSEMBLE DES ARBRES ETIQUETES

1.1.2.1- ARBRS, OCCURRENCES, SOUS-ARBRES.

Soit N_+ l'ensemble des entiers , N_+^* le monoide libre engendré et ε le mot vide et . l'opération de concaténation.

DEFINITION 1.5 : Soit une application t de N_+^* dans FUV et dom(t) son domaine de définition, c'est-à-dire l'ensemble des m appartenant à N_+^* tels que t(m) soit défini. Alors t est un arbre sur FUV si et seulement si

1) ε est élément de dom(t)
2) dom(t) est clos par préfixe i.e.: m appartient à dom(t) si m.m' appartient à dom(t).
3) pour tout m dans dom(t), m.i appartient à dom(t) si et seulement si i est compris entre 1 et l'arité de t(m).

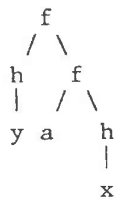
NOTATIONS : Le domaine d'un arbre t est aussi appelé ensemble des occurrences de t. Il sera noté dom(t). Les éléments de dom(t) sont les noeuds de l'arbre, le noeud ε est la racine et m.i est le i-ème fils du noeud m.

Nous ne considérerons dans cette thèse que des arbres finis, c'est-à-dire tels que leurs domaines soient des ensembles finis.

EXEMPLE : Soit F = {f,g,h,a} avec ar(f) = 2, ar(g) = ar(h) = 1 et ar(a) = 0 et V = {x,y}. L'application t définie sur {ε,1,2,11,21,22,221} par:

t(ε)=f , t(1)=h , t(2)=f , t(11)=y , t(21)=a , t(22)=h , t(221)=x

est un arbre que l'on representera graphiquement ainsi:



et qui correspond au terme bien forme f(h(y),f(a,h(x))).

DEFINITION 1.6: Soit t un arbre et m un élément de dom(t). Le sous-arbre de t a l'occurrence m, noté $t_{|m}$ est l'arbre t' défini par t'(m')=t(m.m') pour tout m' appartenant à N_+^*.

EXEMPLE : Dans l'arbre t de l'exemple précédent, le sous-arbre issu du noeud 22 est l'arbre $\begin{array}{c} h \\ | \\ x \end{array}$.

DEFINITION 1.7: Pour tout sous-arbre u de l'arbre t, $t^{-1}(u)$ est l'ensemble des occurrences de u dans t, noté O(u,t). Quand cet ensemble est fini, son cardinal est noté #(u,t).

EXEMPLE : L'ensemble des occurrences de x dans l'arbre t de l'exemple ci-dessus est réduit à {221}.

DEFINITION 1.8: Deux noeuds m et m' sont disjoints si et seulement si il n'existe pas de m" dans N_+^* tels que m=m'.m" ou m'=m.m".

EXEMPLE : Toujours pour le même arbre t, les noeuds 22 et 21 sont disjoints, tandis que 22 et 221 ne le sont pas.

Nous désignerons par M(F,V) l'ensemble des arbres construits sur l'ensemble des symboles F et des variables V.

1.1.2.2- STRUCTURE DE M(F,V)

LEMME 1.1: M(F,V) peut être muni d'une structure de F-algèbre, et c'est la F-algèbre libre engendrée par V.

Nous parlerons donc maintenant indifféremment de termes ou d'arbres.

Afin de pouvoir faire des preuves par récurrence structurelle dans M(F,V), nous allons voir qu'il est possible de construire cet ensemble de facon inductive de la facon suivante:

Soit $M_n(F,V)$ la famille de parties définies par:

-- $M_0(F,V) = V \cup \{c^* \mid c \text{ appartient à } F_0\}$

-- $M_{n+1}(F,V) = M_n(F,V) \cup \{f^*(t_1,\ldots,t_k)$ tel que f appartient à F_k et $t_1,\ldots,t_k$ appartiennent à $M_n(F,V)\}$

LEMME 1.2: M(F,V) est la réunion pour tous les n positifs ou nul des $M_n(F,V)$.

Ce résultat permet de prouver dans M(F,V) des propriétés de la forme: "Pour tout t appartenant à M(F,V), P(t)" par induction sur la structure de t. Il met en évidence le lien existant entre les raisonnements par récurrence structurelle et par récurrence sur les entiers.

Un autre exemple de son utilisation est la définition suivante que l'on peut donner de la taille d'un arbre:

DEFINITION 1.9: La taille d'un arbre t, notée $|t|$, est définie par:

-- si t appartient à VUF_0 alors $|t|=0$

-- si $t=f(t_1,\ldots,t_k)$ alors $|t|=1+\Sigma_{i=1,\ldots k}|t_i|$.

1.1.2.3- OPERATIONS SUR LES ARBRES

Outre les opérations de la F-algèbre et celle qui à un arbre et à une occurrence de son domaine m associe le sous-arbre issu du noeud m, nous en utiliserons une autre qui est le remplacement d'un sous-arbre par un arbre.

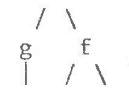
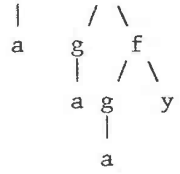
DEFINITION 1.10: Soient t et t' deux arbres et m un noeud de t. Remplacer dans t le sous-arbre $t_{|m}$ par t', c'est définir un nouvel arbre t" noté $t[m \leftarrow t']$ tel que

$$dom(t") = dom(t) - dom(t_{|m}) + m.dom(t')$$

$$t"(m')=t(m') \text{ si } m' \text{ appartient à } dom(t)-dom(t_{|m})$$

$$=t'(m") \text{ si } m'=m.m"$$

t et m étant donnés, on notera t_m l'application de l'ensemble des arbres dans lui-même, qui à un arbre t' associe l'arbre $t_m(t') = t[m\leftarrow t']$.

EXEMPLE : si t_1 =  et t_2 = , $t_1[21 \leftarrow g]$ = 

1.1.2.4- ENDOMORPHISMES DE M(F,V)

DEFINITION 1.11: L'ensemble V(t) des variables d'un arbre t est défini inductivement par:

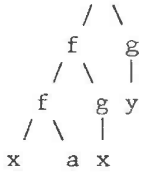
-- si t est une constante alors $V(t) = \emptyset$

-- si t est une variable x alors $V(t) = \{x\}$

-- si $t = f(t_1,\ldots,t_k)$ alors $V(t) = V(t_1) \cup \ldots \cup V(t_k)$

C'est l'ensemble des variables ayant au moins une occurrence dans l'arbre.

NOTATION : Nous désignerons par O(t) l'ensemble des occurrences de t ayant une image dans F, c'est-à-dire l'ensemble des occurrences de non variables.

EXEMPLE : si t =  $V(t) = \{ x, y \}$ $O(t) = \{\varepsilon,1,11,112,12,2\}$

DEFINITION 1.12 : Une substitution est une application σ de V dans M(F,V) telle que $\sigma(x)=x$ presque partout, c'est-à-dire sauf sur un sous-ensemble fini de V appelé le domaine de σ et noté $D(\sigma)$.

$$D(\sigma) = \{ x | \sigma(x) \neq x \}$$

L'ensemble des variables indroduites par σ est noté $I(\sigma)$, il est défini par:

$$I(\sigma) = \bigcup_{x \in D(\sigma)} V(\sigma(x))$$

Le caractère libre de la F-algèbre permet de dire qu'une telle application se prolonge de facon unique en un endomorphisme de M(F,V).
Il vérifiera donc pour tout f d'arité k de F, pour tous $t_1,\ldots,t_k$ de M(F,V):

$$\sigma(f(t_1,\ldots,t_k)) = f(\sigma(t_1),\ldots,\sigma(t_k)).$$

NOTATION : Les substitutions seront désormais désignées par les lettres α , β , α, ... et représentées par leurs graphes, c'est-à-dire par des ensembles de couples $\{(x,\sigma(x))$ pour x dans le domaine de $\sigma\}$, ou bien sous la forme $(x\text{-->}\sigma(x))$.

DEFINITION 1.13 : La composée de deux substitutions ρ et σ est la substitution notée $\rho\ \sigma$ et définie pour toute variable x par

$$\rho\ \sigma\ (x) = \rho\ (\ \sigma(x))$$

DEFINITION 1.14: La restriction d'une substitution σ à un sous-ensemble U de V est notée σ_U et définie par

$$\sigma_U(x) = \sigma(x) \text{ si x appartient à U}$$

$$\sigma(x) = x \quad \text{sinon.}$$

1.1.3- THEORIE EQUATIONNELLE

1.1.3.1- EQUATIONS

DEFINITION 1.15: Une équation est une paire de termes $\{t_1,t_2\}$ notée $(t_1 = t_2)$

DEFINITION 1.16: Une F-algèbre $M = (D_M,F_M)$ valide l'équation $(t_1 = t_2)$ si et seulement si pour tout homomorphisme h de M(F,V) dans D_M, $h(t_1) = h(t_2)$. On dit aussi que l'équation $(t_1 = t_2)$ est valide dans M, et l'on note

$$M \models (t_1 = t_2).$$

Pour déterminer h il suffit de se donner sa restriction h_0 aux variables de V. De plus, comme on ne s'intéresse qu'aux termes t_1 et t_2, il suffit de se donner h_0 sur $V(t_1) \cup V(t_2)$.

DEFINITION 1.17: Une variété de F-algèbres définie équationnellement est une classe de F-algèbres H pour laquelle il existe un ensemble d'équations A tel que H soit la classe de toutes les algèbres validant les équations de A. H est aussi appelée classe des modèles de A et est notée M(A).

G.Birkhoff [BIR,35] a prouvé que si une classe de F-algèbres est une variété, alors la F-algèbre M(A)-libre engendré par V existe pour tout ensemble V.

1.1.3.2- PROBLEME DE VALIDITE . PROBLEME DES MOTS

Etant donné un ensemble d'équations A, le problème consistant à décider si une équation $(t_1=t_2)$ est valide dans toute F-algèbre de M(A) est appelé problème de validité dans M(A) ou problème des mots pour la F-algèbre M(A)-libre. Nous allons voir que ce problème peut s'énoncer en termes de théorie équationnelle.

1.1.3.3- THEORIE EQUATIONNELLE

DEFINITION 1.18: Soit $M=(D_M,F_M)$ une F-algèbre. Une relation R sur D_M est une congruence sur M si et seulement si pour tout symbole de fonction f dans F d'arité n et pour tous éléments $t_1, \ldots, t_n, t'_1, \ldots, t'_n$ dans D_M:

$$t_1 \; R \; t'_1 \;, \ldots, \; t_n \; R \; t'_n \;\Rightarrow f(t_1,\ldots,t_n) \; R \; f(t'_1,\ldots,t'_n).$$

Les opérations de F_M passent au quotient et on peut ainsi définir une algèbre quotient dont le domaine est D_M/R et dont les opérations s'identifient à celle de F_M.

DEFINITION 1.19: Soit A un ensemble d'équations. La plus petite congruence sur M(F,V) contenant toutes les paires $\{ \sigma(t_1) , \sigma(t_2)\}$, où $(t_1=t_2)$ est une équation quelconque de A et σ une substitution, est appelée A-égalité ou congruence engendrée par A et est notée $=_A$.

Deux termes tels que $t_1 =_A t_2$ sont dit A-égaux.

On parlera aussi de la théorie équationnelle engendrée par A pour désigner l'algèbre quotient $M' = (M(F,V)/=_A, F)$.

1.1.3.4- THEOREME DE COMPLETUDE

THEOREME 1.2 [BIR,35] : Etant donné un ensemble d'équations A, une équation $(t_1 = t_2)$ est valide dans la variété M(A) si et seulement si $t_1 =_A t_2$.

Ce théorème dû à G.Birkhoff permet d'étudier le problème de la validité dans la variété M(A) de manière purement syntaxique par l'intermédiaire de la relation $=_A$.

1.1.4- RESOLUTION D'EQUATIONS DANS UNE THEORIE EQUATIONNELLE

Nous allons nous poser maintenant un problème dual du précédent: étant donnés deux termes t_1 et t_2 dans M(F,V) et une algèbre M, existe-t-il un homomorphisme h de M(F,V) dans D_M vérifiant $h(t_1)=h(t_2)$? On dira alors que l'équation $(t_1=t_2)$ est satisfiable dans l'algèbre M. Comme précédemment il suffit de trouver une application h_0 de $V(t_1) \cup V(t_2)$ dans D_M vérifiant la condition requise. De plus, on se restreindra ici au cas où M est l'algèbre M(A)-libre sur V, A étant un ensemble d'équations.

Pour pouvoir décrire l'ensemble des homomorphismes répondant à la question, nous allons définir un système générateur minimal de cet ensemble; cela nécéssite l'utilisation d'un préordre sur les termes et sur les substitutions, obtenu grâce à la notion de filtrage.

1.1.4.1- FILTRE ET PREORDRE DE FILTRAGE

DEFINITION 1.20: La relation $\leq_A$ est définie sur les termes de M(F,V) par:

$$t_1 \leq_A t_2$$

si et seulement si il existe une substitution σ telle que $\sigma(t_1) =_A t_2$.
Si σ existe, σ restreinte à $V(t_1)$ est appelé A-filtre de t_1 vers t_2.

La relation $\leq_A$ est un préordre sur M(F,V), compatible avec la congruence $=_A$.

NOTATION : Dans le cas où A est vide, on notera le préordre de filtrage par $\leq$.

Dans le cas où A est vide, si $t_1 \leq t_2$ on dira que t_1 généralise t_2.
Le filtre de t_1 vers t_2, s'il existe, est alors unique et peut être trouvé par un algorithme récursif simple (voir par exemple [HUE,76]).

EXEMPLE : avec A vide, le terme t_1=f(f(x),y) généralise le terme t_2=f(f(g(a),g(x)), le filtre de t_1 vers t_2 étant la substitution de graphe {(x,g(x)) , (y,g(a)) }.

Un préordre et une relation d'équivalence sur les substitutions se déduisent de la relation $\leq_A$, généralisant les définitions données par G.Huet dans le cas où A est vide [HUE,76].

DEFINITION 1.21: Soient X un ensemble de variables, σ et μ deux substitutions.

$$\sigma \leq_A \mu \ [X]$$

si et seulement si il existe une substitution ρ telle que pour tout x appartenant à X, $(\rho(\sigma(x)) =_A \mu(x)$.

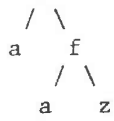
$\sigma \simeq_A \mu$ [X] si et seulement si $\sigma \leq_A \mu$ et $\mu \leq_A \sigma$.

EXEMPLE : dans le cas où A est vide et où X = {x}
σ = { (x,f(a,f(y,b))) } $\leq$ μ = { (x,f(a,f(a,b))) } en prenant ρ = { (y,a) }
et $\sigma \simeq_\emptyset \alpha$ = { (x,f(a,f(a,z))) }.
Si A = { f(a,f(a,x)) = x } et si X = {x}
σ = { (x,f(a,f(y,b))) } $\leq_A \mu$= { (x,b) } en prenant ρ = {(y,a)}.

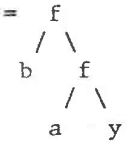
1.1.4.2- UNIFICATEURS DANS UNE THEORIE EQUATIONNELLE

DEFINITION 1.22: Un A-unificateur de deux termes t_1 et t_2 est une substitution σ telle que $\sigma(t_1) =_A \sigma(t_2)$. Si σ existe, t_1 et t_2 sont dits A-unifiables et σ est une solution de l'équation $(t_1 = t_2)$. On note $U(t_1,t_2)$ l'ensemble des A-unificateurs de t_1 et t_2. Un A-unificateur de t_1 et t_2 sera aussi appelé A-solution de $(t_1 = t_2)$.

EXEMPLE : supposons que A = { f = z }

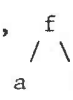


et que t_1 =



et t_2 =

La substitution {(x,b),(y,



)} est un A-unificateur de t_1 et t_2.

1.1.4.3- ENSEMBLE MINIMAL COMPLET DE A-UNIFICATEURS

La relation de préordre $\leq_A$ permet de définir une "base" de l'ensemble $U(t_1,t_2)$, c'est-à-dire un ensemble minimal de A-unificateurs à partir desquels peuvent se déduire tous les autres, par instanciation. Plus précisément:

DEFINITION 1.23: Soient t_1 et t_2 deux termes, $X = V(t_1) \cup V(t_2)$ et W un ensemble fini de variables contenant X. Un ensemble de substitutions Σ est un ensemble complet de A-unificateurs de t_1 et t_2 en dehors de W si et seulement si:

1) $\forall \sigma \in \Sigma \quad D(\sigma) \subset X$ et $I(\sigma) \cap W = \emptyset$

2) $\Sigma \subseteq U(_1,t_2)$

3) pour tout A-unificateur μ, il existe σ dans Σ tel que $\sigma \leq_A \mu$ [X]

La condition 2 traduit la cohérence, la condition 3 la complétude. La condition 1 n'est qu'une condition technique. Elle permet d'éviter les conflits entre les "anciennes" variables et les variables introduites par σ. Nous l'utiliserons uniquement lorsqu'elle apporte de réelles facilités dans les raisonnements.

Si de plus on souhaite que l'ensemble soit minimal, on introduit la condition supplémentaire suivante:

4) pour toutes σ et ρ appartenant à Σ, $\sigma \neq \rho$ implique $\sigma \not\leq_A \rho$ [X].

NOTATION : Un ensemble complet de A-unificateurs de t_1 et t_2 sera noté $ECU_A(t_1,t_2)$. Si de plus il est minimal, on le notera $ECMU_A(t_1,t_2)$.

REMARQUE : -- Pour deux termes donnés, un $ECMU_A$, quand il existe, n'est pas unique: ainsi deux A-unificateurs σ et ρ appartenant à des ECU_A et vérifiant $\sigma =_A \rho$ [X] n'appartiennent pas au même ensemble minimal complet de A-unificateurs.

Rappelons quelques résultats. Dans le cas où A est réduit à l'axiome d'associativité pour un symbole de fonction f et F={f}, G.Plotkin [PLO,72] a décrit un algorithme permettant d'engendrer un ensemble complet minimal de A-unificateurs pour tous termes t et t'. ME.Stickel [STI,81] a donné un algorithme construisant un ensemble minimal complet de A-unificateurs dans le cas où A est constitué des axiomes de commutativité et d'associativité pour un seul symbole de fonction f et pour F={f}. J.Siekmann [SIE,78] a traité le cas de F quelconque, A étant composé des axiomes de commutativité pour un nombre fini de symboles de fonctions. Dans d'autres cas, par exemple pour la structure de groupe abélien [LAN,79], on sait calculer un ensemble complet et fini de A-unificateurs. Nous développerons dans la suite une méthode générale, initialement due à DS.Lankford [LAN,75], reprise par M.Fay [FAY,78], [FAY,79] et améliorée par JM.Hullot

[HUL,80], pour construire des ensembles complets de A-unificateurs dans certaines théories équationnelles. En général ces ensembles ne seront pas minimaux.

1.1.4.4- UNIFICATION DANS L'ENSEMBLE DES TERMES

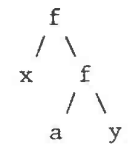
Le cas particulier où il n'y a pas d'équations sur les termes est particulièrement important: en effet dans ce cas, soit les deux termes donnés ne sont pas unifiables, soit il existe un ensemble complet minimal d'unificateurs réduit à un seul élément. Nous allons maintenant donner dans ce cas quelques résultats supplémentaires.

Considérons tout d'abord la relation notée $\simeq$ définie par:

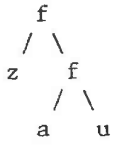
$$t_1 \simeq t_2 \iff t_1 \leq t_2 \text{ et } t_1 \leq t_2.$$

Nous pouvons remarquer qu'alors t_1 et t_2 sont égaux à un renommage de leurs variables près.

EXEMPLE : t_1 =



$\simeq$ t_2 =

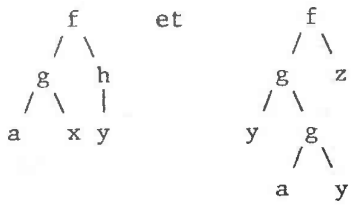


Le théorème fondamental dont la preuve peut être trouvée dans [HUE,76], est alors le suivant :

THEOREME 1.3: Soit Q l'ensemble quotient de l'ensemble des termes par la relation d'équivalence $\simeq$, complété par un élément maximum; alors Q est un treillis complet sans chaine infinie descendante.

On en déduit alors:

PROPOSITION 1.1: Si deux termes t_1 et t_2 sont unifiables, ils admettent un unificateur minimal pour le préordre de filtrage $\leq$ sur les substitutions, unique modulo l'équivalence $\simeq$, et appelé unificateur minimum.

EXEMPLE :  et

ont pour unificateur minimum σ = { (x, ),(y,a),(z,h) }

De nombreux algorithmes d'unification ont été décrits pour trouver cet unificateur minimum. Citons entre autre ceux de JA.Robinson [ROB,65], G.Huet [HUE,76], A.Martelli et U.Montanari [M&M,79], MS.Paterson et MN.Wegmann [P&W,78]. Dans sa thèse, JM.Hullot donne une comparaison de ces différents algorithmes [HUL,80].

1.1.4.5- ENSEMBLE MINIMAL COMPLET DE A-FILTRES

De même que nous avons défini une notion de base de l'ensemble des A-unificateurs de deux termes donnés, nous allons définir, pour deux termes t_1 et t_2 et l'ensemble que nous noterons $F(t_1,t_2)$ de tous les filtres de t_1 vers t_2, une notion analogue.

DEFINITION 1.24: Soient t_1 et t_2 deux termes et $X=V(t_1)$. Un ensemble de substitutions Σ est un ensemble complet de filtres de t_1 vers t_2 si et seulement si:

1) $\forall\ \sigma \in \Sigma,\ D(\sigma) \subset X$

2) Σ est inclus dans $F(t_1,t_2)$

3) $\forall\ \rho \in F(t_1,t_2)\ \exists\ \sigma \in \Sigma$ tel que $\sigma \leq_A \rho\ [X]$

Si de plus Σ vérifie la condition suivante, on dira que c'est un ensemble complet minimal de filtres de t_1 vers t_2.

4) $\forall\ \sigma,\ \sigma' \in F(t_1,t_2)$, si $\sigma \neq \sigma'$, alors $\sigma \not\leq_A \sigma'\ [X]$.

1.1.4.6- EQUATIONS EQUIVALENTES

Nous allons faire dans cette thèse de nombreux raisonnements purement équationnels et manipuler des équations; une notion importante est celle d'équations A-équivalentes, c'est-à-dire ayant même ensemble de A-solutions.

DEFINITION 1.25: Deux équations $(t_1=t_2)$ et $(t'_1=t'_2)$ sont A-équivalentes si et seulement si elles ont des ensembles de A-unificateurs identiques:

$$U(t_1,t_2) = U(t'_1,t'_2)$$

Tout naturellement se pose alors la question suivante: quelle relation existe-t-il entre des ensembles complet minimaux de A-unificateurs de t_1, t_2 et t'_1, t'_2 en supposant qu'ils existent?

PROPOSITION 1.2: Soient $(t_1=t_2)$ et $(t'_1=t'_2)$ deux équations, S un ensemble minimal complet de A-unificateurs des termes t_1 et t_2, $X=V(t_1) \cup V(t_2)$, $Y=V(t'_1) \cup V(t'_2)$ et $S'=\{ \alpha = \sigma_{X \cap Y}$ telle que σ appartienne à S $\}$.
Alors les deux équations $(t_1=t_2)$ et $(t'_1=t'_2)$ sont A-équivalentes si et seulement si S' est un ensemble minimal complet d'unificateurs à la fois de t_1 et t_2 d'une part et de t'_1 et t'_2 d'autre part.

Preuve : il est clair que deux équations ayant un ensemble minimal complet de A-unificateurs commun sont A-équivalentes.
Il faut donc prouver que si les deux équations sont A-équivalentes, S' vérifie la condition requise.
Puisque $(S')' =\{\alpha_{X \cap Y}$ telle que α appartienne à S' $\}$, il suffit de prouver que S' est un ensemble minimal complet de A-unificateurs de t'_1 et t'_2.
Remaquons d'abord que si σ est un A-unificateur de t_1 et t_2, il en est de même pour σ_X qui appartient donc aussi a $U(t'_1,t'_2)$. Donc $\sigma_{X \cap Y}$ appartient $U(t'_1,t'_2)$ et par conséquent aussi à $U(t_1,t_2)$. Le même raisonnement est valable en partant d'une substitution appartenant à $U(t'_1,t'_2)$.

De cette remarque on déduit aisément que S' est inclus dans $U(t'_1,t'_2)$ ce qui est la première condition à remplir pour un ensemble minimal complet de A-unificateurs de t'_1 et t'_2.

Prouvons maintenant la complétude de l'ensemble: si θ est un A-unificateur de t'_1 et t'_2, $\theta_{X \cap Y}$ appartient à $U(t_1,t_2)$.

Il existe donc σ dans S telle que $\sigma_{X} \leq_A \theta_{X \cap Y}$, ce qui implique:

$$\sigma_{X \cap Y} \leq_A \sigma_X \leq_A \theta_{X \cap Y} \leq_A \theta_Y$$

En posant $\alpha = \sigma_{X \cap Y}$, on en déduit que $\alpha_Y \leq_A \theta_Y$.

La condition de minimalité résulte du raisonnement suivant:

posons $\alpha = \sigma_{X \cap Y}$ et $\alpha' = \sigma'_{X \cap Y}$ avec σ et σ' appartenant à S et supposons que $\alpha_Y \leq_A \alpha'_Y$. Comme α et α' sont égales à leurs restrictions sur X et sur Y, nous avons aussi l'inégalité $\alpha_X \leq_A \alpha'_X$, et de plus, par définition de α'

$$\alpha'_X = \alpha' = \sigma'_{X \cap Y} \leq_A \sigma'_X.$$

Il nous reste à prouver que $\sigma_X \leq_A \alpha_X$ pour obtenir la chaine:

$$\sigma_X \leq_A \alpha_X \leq_A \alpha'_X \leq_A \sigma'_X.$$

Ceci vient du fait que, puisque σ appartient à S, α appartient à $U(t_1,t_2)$ et donc il existe θ dans S tel que $\theta_X \leq_A \alpha_X$;

d'où $\theta_X \leq_A \alpha_X \leq_A \sigma_X$.

S étant minimal, θ et σ lui appartenant toutes les deux, elles sont égales. Nous avons ainsi prouvé que

$$\sigma_X \leq_A \alpha_X \text{ et que } \sigma_X \leq_A \sigma'_X$$

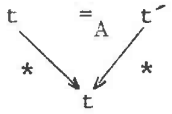
Puisque de même σ et σ' appartiennent à S, elles sont égales, ce qui implique l'égalité de α et α'. []

En supposant que l'on soit capable de calculer et de comparer les ensembles S' de deux équations, ce résultat fournit une procédure de décision pour leur équivalence. Il montre surtout que la définition d'équations équivalentes aurait pu être basée sur le concept d'ensemble minimal complet d'unificateurs, ce qui n'apparait pas de manière évidente dans la définition d'un tel ensemble.

1.2- SYSTEMES DE REECRITURE

Le problème de l'égalité dans une théorie équationnelle est en général indécidable : cependant nous allons développer maintenant une méthode permettant de résoudre ce problème dans le cas d'un grand nombre de théories classiques.

L'égalité dans une théorie équationnelle étant définie par la congruence $=_A$ engendrée par l'ensemble A d'équations, on cherche à déterminer pour chaque classe d'équivalence un représentant canonique et à ramener le problème de la A-égalité de deux termes t_1 et t_2 à celui de l'identité de leurs représentants canoniques. Le processus est le suivant: on oriente les équations de A, ce qui revient à remplacer la congruence $=_A$ par une relation de réduction ---> qui n'est plus symétrique. Les équations orientées constituent alors un système de réécriture de termes. La première condition à imposer est la terminaison du processus, ce qui assure l'existence pour chaque terme t d'une forme normale, c'est-à-dire d'un terme qui n'est plus réductible par ---> et qui s'obtienne à partir de t par un nombre fini d'applications de la relation de réduction. La deuxième condition à exiger pour que cette forme normale soit canonique, est que deux termes équivalents dans la théorie aient même forme normale. Cette condition est connue sous le nom de propriété de Church-Rosser et se visualise de la facon suivante



où $\xrightarrow{*}$ désigne un nombre éventuellement nul d'applications de la relation ---> . Nous étudions en détail ces deux conditions dans ce paragraphe.

Dans le cas où la relation de réduction n'a pas la propriété de terminaison finie, cette méthode ne s'applique plus. Ainsi, il est impossible d'orienter des axiomes de commutativité tout en ayant la terminaison finie du

système de réécriture. Pour contourner cette difficulté, il faut considérer des réductions agissant non plus sur des termes mais sur des classes d'équivalence de termes. Ce sera le sujet d'un paragraphe ultérieur.

1.2.1- SYSTEMES DE REECRITURE DE TERMES

DEFINITION 1.26: On appelle système de réécriture de termes tout ensemble R de couples de termes (g,d) tels que V(d) soit inclus dans V(g). Les éléments de R sont appelés règles de réécriture et notés : g ---> d

A un système de réécriture R, on associe une relation binaire $--->^R$ appelée relation de réduction de la facon suivante:

DEFINITION 1.27: Soient t_1 et t_2 deux termes; t_1 se réduit en t_2 à l'occurrence m de t_1 si et seulement si il existe:

-- une règle (g--->d) de R,

-- une occurrence m dans $D(t_1)$

-- une substitution σ telle que $\sigma(g) = (t_1)_{|m}$

et que $t_2 = t_1[m <- \sigma(d)]$.

On note alors $t_1 --->^R t_2$ et on dit que t_1 est réductible ou que t_1 se réécrit en t_2.

Ainsi un terme est réductible si l'un de ses sous-termes est l'image par une certaine substitution d'un membre gauche de règle. Le terme réduit est obtenu en remplacant le sous-terme en question par l'image dans la même substitution du membre droit de la règle.

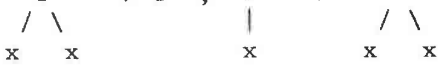
NOTATIONS : ** Dans la suite on notera ---> au lieu de $--->^R$ quand il n'y a pas d'ambiguité sur le système de réécriture R.

** $\xrightarrow{+}$ désignera la fermeture transitive de la relation --->,

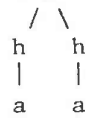
** $\xrightarrow{*}$ sa fermeture réflexive transitive et

** <---> sa fermeture symétrique.

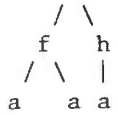
EXEMPLE : Soit R = { f ---> b , h ---> f }



Le terme t = f se réduit en b à l'occurrence ε.



Mais il se réduit aussi en f à l'occurrence 1.



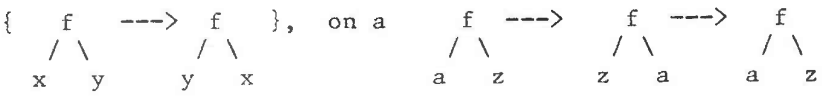
DEFINITION 1.28: Un terme t est dit en forme normale ou irréductible si et seulement si il n'existe pas de terme t' tel que $t \xrightarrow{R} t'$, c'est-à-dire si t ne peut plus se réécrire dans R.

DEFINITION 1.29: Si t et t' sont deux termes tels que t' soit irréductible et $t \xrightarrow{*}{}^{R} t'$, t' est appelé une forme normale de t et on la note FN(t).

EXEMPLE : en reprenant l'exemple précédent, b est une forme normale de t.

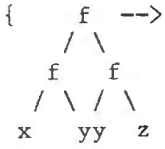
Sans l'hypothèse de terminaison finie du processus de réécriture, l'existence d'une forme normale n'est en aucun cas assurée: par exemple si

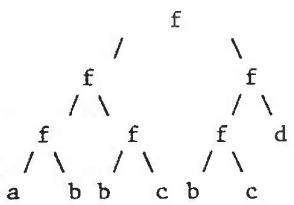
R = { f ---> f }, on a



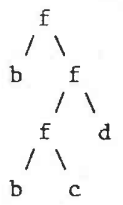
D'autre part, si un terme a une forme normale, elle n'est pas en général unique:

par exemple si R = { f --> y } le terme





se réécrit à l'occurrence 1 en



irréductible,

mais aussi à l'occurrence ϵ en

également irréductible.

Nous définissons aussi une notion de forme normale sur les substitutions:

DEFINITION 1.30: Une substitution σ est normalisée si et seulement si pour toute variable x de son domaine $D(\sigma)$, $\sigma(x)$ est en forme normale.

1.2.2- TERMINAISON D'UN SYSTEME DE REECRITURE

DEFINITION 1.31: Un système de réécriture R est noethérien si et seulement si pour tout terme t, il n'existe pas de dérivation infinie $t = t_0 \dashrightarrow t_1 \dashrightarrow \ldots$ On dit aussi dans ce cas que la relation ---> a la propriété de terminaison finie.

Rappelons que quand un système est noethérien, tout terme a au moins une forme normale.

Nous ne nous étendrons pas sur les diverses méthodes proposées pour prouver la terminaison des systèmes de réécriture, mais nous allons expliciter celle que nous emploierons dans la suite et qui est due à N.Dershowitz [DER,79] Citons par ailleurs les travaux de P.Lescanne, F.Reinig, J.P.Jouannaud et H.Kirchner, S.Kamin et J.J.Levy, G.Choque, sur le sujet ([LES,81], [REI,81], [J&K,82], [JLR,82], [K&L,82], [CHO,82]).

DEFINITION 1.32: Un ordre partiel strict $<$ sur l'ensemble des termes est un ordre de simplification si et seulement si il possède les trois propriétés suivantes:

pour tout symbole de fonction f dans F, pour tous termes t et t',

1) $t > t'$ implique $f(...t...) > f(...t'...)$

2) $f(...t...) > t$

3) $f(...t...) > f(... ...)$

Les points de suspension signifient que les autres sous-termes restent inchangés.

La condition 1 est la propriété de compatibilité avec la structure des termes, la condition 2 est appelée propriété des sous-termes, la condition 3 est une propriété d'effacement utile si certains opérateurs n'ont pas une arité fixe, et nous ne l'utiliserons pas dans la suite.

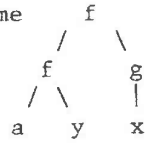
Un exemple particulièrement important d'ordre de simplification est l'ordre de plongement défini de la facon suivante:

DEFINITION 1.33: On dit que le terme $t=f(t_1,...,t_n)$ est plongé dans le terme $t'=g(t'_1,...,t'_p)$ et on note $t \underline{\Delta} t'$, si et seulement si l'une des conditions suivantes est verifiée:

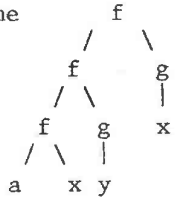
1) il existe i tel que t soit plongé dans t'_i

2) $f=g$ et pour tout i , t_i est plongé dans t'_i

EXEMPLE : le terme



est plongé dans le terme



L'importance de l'ordre de plongement vient d'une part du fait que tout autre ordre de simplification le contient, c'est-à-dire:

PROOSITION 1.3 : Tout ordre de simplification $<$ contient l'ordre de plongement.

$$\forall\ t, t' \in M(F,V),\ t \underline{\Delta}\ t' \Rightarrow t < t'$$

Et d'autre part cet ordre intervient de facon déterminante dans l'établissement d'une condition suffisante de terminaison d'un système de réécriture:

PROPOSITION 1.4: Soit R un système de réécriture de termes tel que le nombre de symboles de fonctions apparaissant dans R soit fini. Alors R termine si il existe un ordre de simplification $<$ sur M(F,V) tel que, pour toute substitution σ et pour toute règle g--->d dans R, on ait σ (g) $>$ σ (d).

La preuve de ce résultat s'appuie sur les propriétés du plongement et sur le théorème de Kruskal [KRU,60]. Elle est assez complexe et peut être trouvée dans [DER,79].

Pour construire un ordre de simplification sur les termes, nous utiliserons l'ordre récursif sur les chemins ("recursive path ordering"), qui est construit à partir d'un ordre partiel sur les symboles de fonctions. Cette méthode a été décrite par D. Plaisted [PLA,78], reprise et améliorée par N. Dershowitz [DER,79].

Pour définir l'ordre récursif sur les chemins nous utiliserons la notion de multi-ensemble sur un ensemble P et l'extension naturelle d'un ordre sur P aux multi-ensembles sur P (voir [JLR,82]).

DEFINITION 1.34: Un multi-ensemble S sur un ensemble P est une application de P dans l'ensemble des entiers naturels N.

EXEMPLE : {1,3,3,5,5,5} est un multi-ensemble sur N correspondant à l'application S définie par S(1)=1, S(3)=2, S(5)=3, S(x)=0 pour tout autre entier x.

DEFINITION 1.35: Etant donné un ordre strict $<_p$ sur P, on lui associe l'ordre $<<_p$ sur les multi-ensembles sur P de la facon suivante:

$S <<_p S'$ si et seulement si S est différent de S' et

pour tout élément x de P, si $S(x) < S'(x)$ alors il existe un autre élément y de P tel que $x <_p y$ et $S(y) < S'(y)$.

De facon plus intuitive, si un élément x a plus d'occurrences dans S que dans S', il existe un élément y plus grand que x, ayant plus d'occurrences dans S' que dans S.

EXEMPLE : $\{1,3,3,5,5,5\} <<_N \{1,1,3,6\}$ puisque par exemple $S(5)=3$ mais $5<6$ et $S(6)=0$ est strictement inférieur à $S'(6)=1$. Il suffit de faire le même raisonnement pour 3 et de remarquer que $S(1) < S'(1)$.

DEFINITION 1.36: Etant donné un ordre partiel $<$ sur l'ensemble des symboles de fonctions F, l'ordre récursif sur les chemins est ainsi défini sur l'ensemble des termes:

$$s=f(\ldots,s_i,\ldots) \overset{*}{<} t=g(\ldots,t_i,\ldots)$$

si et seulement si l'une des conditions suivantes est verifiée:

1) $f=g$ et $\{\ldots,s_i,\ldots\} \overset{**}{<<} \{\ldots,t_i,\ldots\}$
2) $f<g$ et pour tout i, $s_i \overset{*}{<} t$
3) $f \not< g$ et il existe j tel que $s \overset{*}{<} t_j$ ou $s = t_j$.

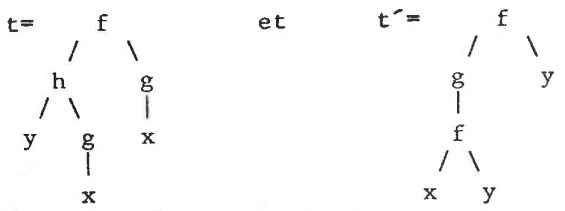
Le résultat fondamental prouvé par N.Dershowitz est le suivant:

PROPOSITION 1.5 [DER,79]: $\overset{*}{<}$ est un ordre de simplification.

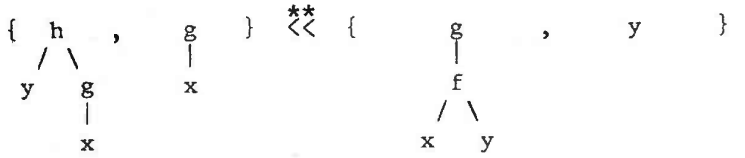
La preuve en est longue et technique. Montrons sur un exemple comment comparer deux termes.

EXEMPLE : supposons que F={f, g, h} et que g<h<f.

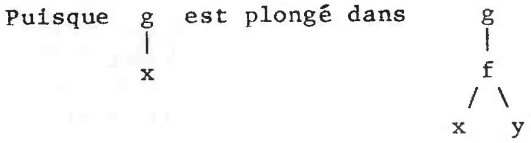
Montrons que les termes



En effet t $\overset{*}{<}$ t´ <=> (par le cas 1, puisque les symboles de fonctions à l´occurrence ε sont les mêmes)

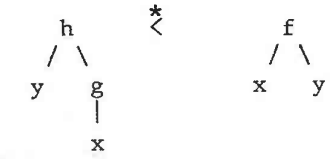


Puisque g est plongé dans g , il est nécessaire et suffisant de prouver

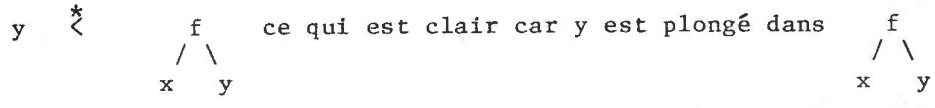


que h $\overset{*}{<}$ g

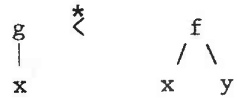
<=> (par le cas 3, puisque h>g implique h $\not\leq$ g)



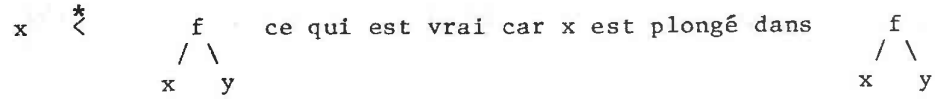
<=> (par le cas 2, puisque h<f)



et



<=> (par le cas 2, puisque g<f)



Un intérêt fondamental de l'ordre récursif sur les chemins $\overset{*}{<}$ déduit de l'ordre $<$ est qu'au lieu de vérifier, pour toute règle g--->d de R et pour toute substitution σ , que $\sigma(d) \overset{*}{<} \sigma(g)$, il suffit de montrer que $d \overset{*}{<} g$ pour toute règle de R. Cela résulte du théorème suivant montrant que l'ordre récursif sur les chemins est stable par instanciation.

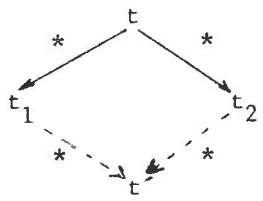
PROPOSITION 1.6: Etant donné un ordre $<$ sur F, quels que soient le termes t et t', si $t \overset{*}{<} t'$, alors pour toute substitution σ, $\sigma(t) \overset{*}{<} \sigma(t')$.

1.2.3- CONFLUENCE D'UN SYSTEME DE REECRITURE

1.2.3.1- CONFLUENCE

DEFINITION 1.37: Un système de réécriture est confluent si et seulement si pour tous termes t, t_1, t_2 tels que $t \overset{*}{\longrightarrow} t_1$ et $t \overset{*}{\longrightarrow} t_2$, il existe un terme t' tel que $t_1 \overset{*}{\longrightarrow} t'$ et $t_2 \overset{*}{\longrightarrow} t'$.

On l'exprime par le diagramme suivant:



REMARQUES : -- Il est facile de prouver que la confluence et la propriété de Church-Rosser sont des propriétés équivalentes.

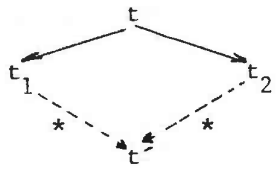
-- Quand un système de réécriture est confluent, tout terme a au plus une forme normale.

La confluence d'un système de réécriture est en général indécidable. Mais ce n'est pas le cas pour des systèmes noethériens, comme nous allons le voir maintenant. L'idée de base est de localiser le test de confluence.

1.2.3.2- LOCALE CONFLUENCE ET LEMME DU LOSANGE

DEFINITION 1.38: Un système de réécriture est localement confluent si et seulement si pour tous termes t, t_1, t_2, tels que $t \longrightarrow t_1$ et $t \longrightarrow t_2$, il existe un terme t' tel que $t_1 \overset{*}{\longrightarrow} t'$ et $t_2 \overset{*}{\longrightarrow} t'$.

Cette propriété, dite propriété du losange, se visualise par le diagramme:



LEMME DU LOSANGE 1.3 : Un système de réécriture noethérien est confluent si et seulement si il est localement confluent.

Ce lemme, dû à M.Newmann [NEW,42], doit son importance au fait qu'il existe un moyen simple de tester la confluence locale d'un système de réécriture. Ce test, que nous allons décrire maintenant, est dû à D.Knuth et P.Bendix.

1.2.3.3- PAIRES CRITIQUES ET THEOREME DE KNUTH-BENDIX

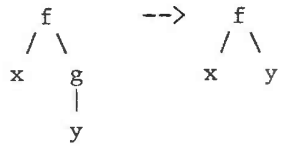
Nous supposerons dans toute la suite que deux règles g--->d et g'--->d' de R vérifient $V(g) \cap V(g') = \emptyset$, condition qui peut toujours être réalisée par un renommage des variables.

DEFINITION 1.39: Soient (g,d) et (g',d') deux couples de termes. Supposons qu'il existe un sous-terme non variable de g unifiable avec g'. Soient m l'occurrence de ce sous-terme et σ l'unificateur minimum de $g_{|m}$ et de g'.
Alors la paire $\{ \sigma(g[m \leftarrow d']), \sigma(d) \}$ est appelée paire critique, obtenue par superposition de (g',d') sur (g,d).

DEFINITION 1.40: On appelle paire critique de R, toute paire critique obtenue par superposition de (g´,d´) sur (g,d) pour tout choix possible des règles (g--->d) et (g´--->d´) dans R.

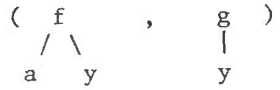
On n´a pas supposé dans cette définition les deux règles distinctes; on peut en effet superposer une règle sur elle-même, en remarquant toutefois que la superposition à l´occurrence ε fournit une paire critique trivialement confluente.

EXEMPLE : en superposant la règle



sur la règle

à l´ occurrence ε , on obtient la paire critique:



L´intérêt des paires critiques vient du théorème suivant:

THEOREME DE KNUTH-BENDIX 1.4: Un système de réécriture de termes R est localement confluent si et seulement si pour toute paire critique (t, t´) de R, il existe t" tel que $t \xrightarrow{*} t"$ et $t' \xrightarrow{*} t"$.

Ce théorème a tout d´abord été prouvé par D.Knuth et P.Bendix [K&B,70], en utilisant l´hypothèse de terminaison finie de la relation de réécriture. G.Huet [HUE,80] en a donné une preuve sans cette hypothèse.

Dans le cas où le système R est fini et noethérien, ce résultat donne une procédure de décision pour la confluence de R: comme il n´existe qu´un nombre fini de paires critiques, il suffit pour chacune d´elles de calculer les formes normales des deux membres et de tester l´égalité.

1.2.3.4- SYSTEME COMPLET OU CANONIQUE

DEFINITION 1.41: Un système de réécriture à la fois noethérien et confluent est dit système complet ou canonique.

Si R est un système complet, tout terme t admet une unique forme normale. En notant $=_R$ la fermeture réflexive symétrique transitive de la relation de réduction $--->^R$, il est équivalent de prouver pour deux termes t et t', $t =_R t'$ et FN(t) = FN(t').

La question que nous nous proposons d'étudier maintenant est de trouver, quand c'est possible, un système de réécriture complet déduit d'un ensemble d' équations A, tel que les congruences $=_A$ et $=_R$ coincident. Ainsi l'identité des formes normales fournira un processus de décision de la A-égalité.

1.2.3.5- ALGORITHME DE COMPLETION

Considérons un ensemble d'axiomes A et supposons qu'il existe un ordre de simplification <, tel que, pour toute équation (g=d) dans A, on ait soit g<d, soit d<g.

A toute équation g=d, on associe alors la règle de réécriture

g ---> d si g>d

d ---> g si d>g.

Soit Rl le système de réécriture ainsi obtenu: si Rl est complet, c'est-à-dire si toute paire critique (t, t') de Rl vérifie FN(t)=FN(t'), alors les congruences $=_A$ et $=_R$ coincident.

Dans le cas contraire, désignons par A1 l'ensemble des paires (FN(t), FN(t')) telles que (t, t') soit une paire critique de R1 dont les formes normales des deux membres soient différentes. On va tenter de rendre R1 localement confluent en introduisant les équations de A1 comme nouvelles règles de réécriture, tout en gardant la propriété de terminaison finie, de manière à garantir le calcul des formes normales et la confluence éventuelle du nouveau système.

Ce processus est dû à D.Knuth et P.Bendix [K&B,70]. La preuve de la correction de l'algorithme a été donnée par G.Huet [HUE,82]. Dans sa thèse, J.M.Hullot [HUL,80] reprend cet algorithme et en donne plusieurs versions. Il en a d'autre part fait une implémentation en VLISP, qui remaniée et complétée constitue une partie du système FORMEL, réalisé sur le système MULTICS par G.Huet et son équipe à l'INRIA.

En nous inspirant du travail de J.M.Hullot, nous donnons maintenant deux versions structurées de l'algorithme de complétion. La première est très proche de celle décrite par Knuth et Bendix.

Première version de l'algorithme de complétion

Initialisation: R <- ∅, E <- A, E' <- ∅.

TANT QUE vrai FAIRE

Réduction des paires:

E'<- {(FN(t), FN(t')) / (t, t') appartient à E et FN(t)≠FN(t') }

SI E' est vide ALORS arrêt succès FSI

R'<- ∅

TANT QUE E' est non vide FAIRE

choix d'une équation (FN(t)=FN(t')) dans E':

E'<- E'-{(FN(t), FN(t'))}

orientation de cette équation:

SI FN(t) et FN(t') non comparables pour <, ALORS arrêt échec

SINON

SI FN(t) > FN(t') ALORS g <- FN(t), d <- FN(t')

SINON g <- FN(t'), d <- FN(t)

FIN SI

R'<- R' U {g--> d}

FIN SI

FIN TANT QUE

Ajouter les nouvelles règles:

R <- R U R'

Calcul des paires critiques:

E <- { paires critiques calculées par superposition des règles de R' sur celles de R d'une part, et par superposition des règles de R' entre elles d'autre part }

FIN TANT QUE

Faisons d'abord quelques commentaires sur cette première version.

Trois situations peuvent se produre:

* soit le processus s'arrête avec succès, au bout d'un nombre fini d'étapes, en retournant un système R canonique. Mais rien ne dit que ce soit le seul possible.

* soit il s'arrête en échec, car l'ordre $<$ choisi ne permet pas d'orienter toutes les équations de A. Cependant, cela ne signifie pas qu'il n'existe pas de système complet.

* soit il ne s'arrête jamais, générant un système infini de règles. Même dans ce cas, il se peut qu'un système complet existe, nous en verrons un exemple dans la suite de cette thèse.

De plus, ce processus n'est pas satisfaisant d'un autre point de vue. En effet, lors de l'introduction de nouvelles règles, des réductions peuvent apparaitre entre les différentes règles du nouveau système.

Dans une version améliorée par G.Huet de cet algorithme, on garde toujours les règles en forme normale les unes par rapport aux autres; aussi n'aura-t-on plus en général l'inclusion des systèmes de réécriture successifs construits par l'algorithme, puisque certaines règles pourront être supprimées lors de l'adjonction de nouvelles règles.

Pour mettre en oeuvre ce nouvel algorithme, il faut:

* numéroter les règles dans leur ordre d'apparition. On notera $g_k \text{--->} d_k$ la règle de numéro k.

* les marquer chaque fois que l'on a calculé toutes les paires critiques de la règle numérotée k avec les règles de numéro k' inférieur ou égal à k.

Deuxième version de l'algorithme de complétion

Initialisation: R<- ∅, p<- 0, E<- A.

TANT QUE vrai FAIRE

TANT QUE E est non vide FAIRE

Réduction des équations:

Choisir une équation (t=t') dans E

Calcul de FN(t) et FN(t') dans R

Orientation de (FN(t), FN(t')):

CAS * FN(t) et FN(t') incomparables pour < ALORS arrêt échec

* FN(t)=FN(t') ALORS E<- E-{(t=t')}

* SINON SI FN(t) > FN(t') ALORS g<- FN(t), d<- FN(t')

SINON g<- FN(t'), d<- FN(t)

FSI

Ajout d'une nouvelle règle:

calculer K={k / g_k--->d_k appartient à R et g_k

se réduit en g'_k par la règle g--->d }

E<- E-{(t=t')} U {(g'_k=d'_k) / k appartient à K et

d_k se réduit en d'_k par la règle g--->d }

p <- p+1

R <- {g_k-->d'_k / k n'appartenant pas à K} U {g--->d}

la nouvelle règle k est marquée si l'ancienne l'était;

la règle g--->d est numérotée p et non marquée.

FIN CAS

FIN TANT QUE

```
        Calcul des paires critiques:
        SI toutes les règles de R sont marquées
                ALORS arrêt succès
                SINON  choisir une règle non  marquée  de  numéro  k;
                       E <- {paires critiques de la règle numéro k avec les
                             règles  de  numéro k' inférieur ou égal à k};
                       marquer la règle k
                FSI
FIN TANT QUE
```

J.M.Hullot décrit dans sa thèse de nombreux exemples d'utilisation de cet algorithme. Le lecteur peu familier avec cette technique peut s'y référer [HUL,80].

1.3- ETUDE D'EXTENSIONS DE LA NOTION DE SYSTEME DE REECRITURE DE TERMES

Nous allons rappeler dans ce paragraphe trois techniques permettant d'étendre la notion de réécriture de termes à des théories que la notion classique de réécriture ne permet pas d'étudier. C'est en particulier le cas dans les théories comportant des axiomes de commutativité ou plus généralement de type permutatif. En effet, on ne sait pas orienter ces axiomes de telle sorte que le système de réécriture associé soit à terminaison finie. Cela peut aussi être le cas lorsque certains axiomes perturbent l'étude d'un processus particulier comme par exemple la résolution d'équations. Nous aurons l'occasion d'y revenir et d'illustrer ce dernier fait.

L'idée centrale de toutes les extensions est de partitionner l'ensemble des axiomes A en deux ensembles E et R, tels que l'on puisse traiter R comme un système de réécriture de termes ayant de "bonnes" propriétés par rapport à E. D'autre part, on exigera des deux membres des équations (g=d) de E de faire intervenir les mêmes ensembles de variables: V(d)=V(g).

La théorie des groupes abéliens constitue un exemple classique dans lequel l'ensemble des équations consiste en la réunion des ensembles E et R, tels que, en notant la loi de composition interne * :

$$E = \{ (x*y = y*x) , (x*(y*z) = (x*y)*z) \}$$

$$R = \{ (x*1=x) , (x*x' = 1) \}$$

De même que précédemment, nous souhaitons disposer d'une notion de représentant canonique d'une classe d'équivalence de $=_A$. Mais ici il nous faudra travailler modulo les équations de E, c'est-à-dire que nous déterminerons une classe modulo E représentant canoniquement une classe modulo A.

Dans tout ce qui suit, nous noterons ---> la relation de réduction associée à R, dont les axiomes seront supposés orientés de gauche à droite.

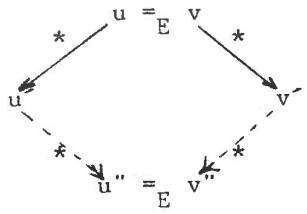
1.3.1- CONFLUENCE MODULO UN ENSEMBLE D'EQUATIONS.

Cette méthode est due à G.Huet [HUE,80]. Sa caractéristique essentielle est de ne pas requérir la connaissance d'un algorithme de E-unification. Cependant on ne pourra considérer que des systèmes de réécriture linéaires à gauche, c'est-à-dire tels que dans chaque membre gauche de règle toute variable n'apparaisse qu'une seule fois.

1.3.1.1- DEFINITIONS

Nous introduisons tout d'abord une nouvelle notion de confluence.

DEFINITION 1.42: Un système de réécriture de termes R est dit confluent modulo E si et seulement si pour tous termes u, v, u', v' on a
$u =_E v$ et $u \xrightarrow{*} u'$ et $v \xrightarrow{*} v'$ => $\exists u"$ et v" tel que $u' \xrightarrow{*} u"$ et $v' \xrightarrow{*} v"$ et $u" =_E v"$
ce qui peut s'exprimer par le diagramme suivant:



Notons que cette propriété n'est pas la confluence de ---> dans l'ensemble des classes, puisqu'on ne s'autorise pas de $=_E$ le long des branches descendantes du diagramme, c'est-à-dire au cours des dérivations.

Afin d'étendre le théorème de Knuth et Bendix, il nous faut définir une nouvelle notion de paire critique:

DEFINITION 1.43: * On appelle paire critique de R/E toute paire critique obtenue par superposition d'une équation sur une règle, c'est-à-dire par superposition

du couple (g,d) avec (g=d) $\in$ E ou (d=g) $\in$ E

sur le couple (g',d') avec (g'--->d') $\in$ R.

* On appelle paire critique de E/R toute paire critique obtenue par superposition d'une règle sur une équation, c'est-à-dire par superposition

du couple (g',d') avec (g'--->d') $\in$ R

sur le couple (g,d) avec (g=d) $\in$ E ou (d=g) $\in$ E.

1.3.1.2- LE THEOREME DE HUET

Nous pouvons maintenant énoncer le théorème étendant le résultat de Knuth et Bendix. Pour le montrer, on est amené à localiser, d'une part la réécriture --->, mais aussi la E-égalité $=_E$. Sa preuve peut être trouvée dans [HUE,80].

THEOREME 1.5: Soit $=_A$ une théorie équationnelle dont l'ensemble des axiomes A peut être partitionné en deux ensembles E et R tels que:

*** R est un système de réécriture de termes linéaire à gauche;

*** pour toute équation (g=d) de E, on a V(d)=V(g);

*** $\xrightarrow{}.=_E$ a la propriété de terminaison finie.

Alors, en notant $FN_R(t)$ la fore normale d'un terme t pour --->, R est confluent modulo E si et seulement si pour toutes les paires critiques (u,v) de R, de R/E, de E/R, on a $FN_R(u) =_E FN_R(v)$.

Dans ce cas, pour tous termes u et v, on a $u =_A v$ si et seulement si $FN_R(u) =_E FN_R(v)$.

REMARQUES : -- La condition V(g)=V(d) pour toute équation de E découle de la condition de terminaison finie de $\xrightarrow{}.=_E$.

-- Il n'est pas nécessaire de tester les paires critiques de E sur E.

-- Dans le cas où R et E sont finis et satisfont les hypothèses du théorème et lorsque la E-égalité est décidable, ce théorème fournit une procédure de décision pour la confluence modulo E, puisque le nombre de paires critiques est alors fini.

-- La condition de linéarité des membres gauches des règles de R est en pratique très restrictive.

On déduit de ce théorème un algorithme de complétion. On pourra consulter à ce sujet [HUL,81], pour sa description et des exemples d'utilisation.

1.3.2- CONFLUENCE DANS LA STRUCTURE QUOTIENT

Comme nous venons de le voir, la méthode précédente, bien que ne nécessitant pas la connaissance d'un algorithme de E-unification, est assez limitative du fait de l'hypothèse de linéarité à gauche faite sur le système de réécriture R. Nous allons ici donner une seconde approche qui permettra, en supposant la connaissance d'un algorithme de E-unification, de se passer de l'hypothèse de linéarité à gauche du système R.

1.3.2.1- GENERALITES

On définit une nouvelle relation notée $--->^{R/E}$ sur l'ensemble des termes par

$$--->^{R/E} = =_E . --->$$

Un terme réductible par $--->^{R/E}$ sera dit R/E-réductible.

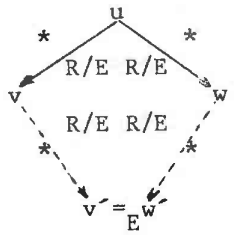
Cette approche est très différente de la précédente puisqu'on s'autorise des étapes de E-égalité au cours des réductions.

DEFINITION 1.44: Le système de réécriture R est dit E-confluent si et seulement si pour tous termes u, v, w, la propriété suivante est vérifiée:

$$\text{si } u \xrightarrow{*}{}^{R/E} v \text{ et } u \xrightarrow{*}{}^{R/E} w$$

alors il existe v' et w' tels que $v \xrightarrow{*}{}^{R/E} v'$, $w \xrightarrow{*}{}^{R/E} w'$ et $v' =_E w'$

ce qui peut s'exprimer graphiquement par le diagramme suivant:



Un système de réécriture de termes R sera dit E-noethérien si et seulement si la relation $--->^{R/E}$ a la propriété de terminaison finie, il sera dit E-canonique si et seulement si il est à la fois E-confluent et E-noethérien.

REMARQUE: La E-confluence de R n'est autre que la confluence de la relation ---> dans la structure quotient $M(F,V)/=_E$. Mais ce n'est pas la confluence de $--->^{R/E}$ puisqu'une dernière étape de E-égalité est autorisée.

Si le système de réécriture R est E-canonique, en désignant par $FN_{R/E}(t)$ une forme normale pour la relation $--->^{R/E}$, on a

$$u =_A v \iff FN_{R/E}(u) =_E FN_{R/E}(v)$$

Pour pouvoir travailler avec la relation $--->^{R/E}$, il faut pouvoir en particulier décider la R/E-réductibilité d'un terme. Cela est possible de facon claire si les classes d'équivalences modulo E sont finies. En général, on est amené à introduire la notion de E-uniformité:

DEFINITION 1.45: Une relation ===> sur l'ensemble des termes M(F,V) sera dite E-uniforme si et seulement si, pour tous termes u et v tels que $u --->^{R/E} v$, il existe un terme w tel que u ===> w.

Ainsi pour une relation E-uniforme, les notions de réductibilité pour $--->^{R/E}$ et pour ===> se confondent.

Une première méthode pour tester si un système de réécriture est E-confluent a été étudiée par G.Huet. Elle est sommairement exposée dans [HUL,81]. Mais, comme dans le cas de la confluence modulo E, si la connaissance d'un algorithme de E-unification n'est pas requis, il faut ici encore supposer le système de réécriture R linéaire à gauche, pour donner un analogue du théorème de Knuth et Bendix.

Nous allons maintenant développer une seconde méthode due à G.E.Peterson et M.E.Stickel [P&S,81].

1.3.2.2- UTILISATION D'UN ALGORITHME DE E-UNIFICATION

Dans cette partie, pour étudier la relation $--->^{R/E}$, nous introduisons une nouvelle relation de réduction étendant --->.

DEFINITION 1.46: Un terme u est R,E-réductible en un terme v à l'occurrence m, et nous noterons $u --->^{R,E} v$, si et seulement si il existe une règle g--->d de de R et une substitution σ telles que

$$u_{|m} =_E \sigma(g) \text{ et } v = u[m <- \sigma(d)]$$

REMARQUE : On a $---> \subseteq --->^{R,E} \subseteq --->^{R/E}$.
La R,E-réductibilité est décidable lorsque R est fini (ce que nous supposerons toujours), si le E-filtrage est décidable. Cela sera le cas en particulier pour des axiomes de commutativité, d'associativité ou encore de commutativité et d'associativité pour le même symbole de fonction, puisqu'il existe dans ces situations un algorithme complet et fini de E-filtrage.

Il est facile de vérifier les propriétés suivantes de la relation $--->^{R,E}$ qui seront utilisées ultérieurement:

* Soient trois termes u, v, t et m une occurrence de dom(t).
Si $u --->^{R,E} v$, alors $t[m<-u] --->^{R,E} t[m<-v]$.

* Pour toute substitution σ et tous termes t et t´,
si $t \xrightarrow{R,E} t'$, alors $\sigma(t) \xrightarrow{R,E} \sigma(t')$.

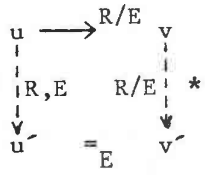
1.3.2.2.1- LA E-COMPATIBILITE

Pour étudier la relation $\xrightarrow{R/E}$ à l´aide de $\xrightarrow{R,E}$ nous sommes amenés à introduire une condition beaucoup plus forte que la E-uniformité, qui est la propriété de E-compatibilité définie par G.E.Peterson et M.E.Stickel. [P&S,81].

DEFINITION 1.47: Nous dirons qu´un système de réécriture de termes R est E-compatible si et seulement si, pour tous termes u et v tels que $u \xrightarrow{R/E} v$, il existe des termes u´ et v´ tels que

$$u \xrightarrow{R,E} u' \text{ et } v \xrightarrow{*}_{R/E} v' \text{ et } u' =_E v'$$

ce que l´on peut exprimer graphiquement par le diagramme suivant:



REMARQUE : Si R est E-compatible, alors la relation $\xrightarrow{R,E}$ est E-uniforme et la R/E-réductibilité se réduit à la R,E-réductibilité.

Afin d´obtenir une condition effectivement testable pour déterminer si un système de réécriture de termes R est E-compatible, on va introduire une nouvelle notion de paire critique associée à la R,E-réécriture.

1.3.2.2.2- PAIRES E-CRITIQUES

Le mécanisme que nous décrivons ci-dessous sera appelé algorithme de E-superposition.

DEFINITION 1.48: Soient (g,d) et (g',d') deux couples de termes tels que $V(g) \cap V(g') = \emptyset$. Supposons qu'il existe un sous terme non variable de g d'occurrence m qui soit E-unifiable avec g' et désignons par Σ un ensemble complet de E-unificateurs de $g_{|m}$ et g' en dehors de $V(g) \cup V(g')$. Alors toutes les paires

$$\{ \sigma(g[m \leftarrow d']), \sigma(d) \} \text{ avec } \sigma \in \Sigma$$

sont appelées paires E-critiques obtenues par E-superposition de (g',d') sur (g,d). On dit qu'elles forment un ensemble complet de paires E-critiques de (g',d') sur (g,d) à l'occurrence m.

NOTATIONS : On dit que PC est un ensemble complet de paires E-critiques de R, si et seulement si PC contient un ensemble complet de paires E-critiques de (g',d') sur (g,d) à l'occurrence m, pour tout choix de m, occurrence de non variable dans g, et pour tout choix possible de deux règles (g'--->d') et (g--->d) de R.

De la même facon que nous avons introduit précédemment les notions de paires critiques de R/E et de E/R, nous allons maintenant définir la notion de paire E-critique de R/E. Cependant on ne superposera pas ici les équations sur les règles:
en effet, la superposition d'une équation (g'=d') de E sur une règle g--->d de R, donne la paire E-critique $\{ \sigma(g[m \leftarrow d']), \sigma(d) \}$. Mais comme
$\sigma(d') =_E \sigma(g') =_E \sigma(g_{|m})$, on a $\sigma(g[m \leftarrow d']) =_E \sigma(g)$,
et par conséquent la partie gauche de cette paire critique est R,E-réductible en sa partie droite. Pour cette raison, nous n'avons pas à considérer de telles paires critiques.

DEFINITION 1.49: On appelle paire E-critique de R/E toute paire E-critique obtenue par l'algorithme de E-superposition appliqué aux couples (g,d) et (g',d') tels que (g=d) ou (d=g) soit élément de E et (g'--->d') appartienne à R. On dit que PC est un ensemble complet de paires E-critiques de R/E si et seulement si PC contient un ensemble complet de paires E-critiques de (g',d') sur (g,d) pour tout choix d'une occurrence m de non variable de g, et pour tout choix d'une équation (g=d) ou (d=g) dans E et d'une règle (g'--->d') dans R.

1.3.2.2.3- UNE CONDITION SUFFISANTE DE E-CONFLUENCE

Nous sommes alors en mesure de donner une condition suffisante de E-compatibilité.

LEMME 1.4: Soit E un ensemble d'équations (g=d) telles que g et d soient linéaires et tels que V(g)=V(d). Alors si pour toute paire E-critique (u,v) de R/E, il existe un terme w tel que

$$u \xrightarrow{*}_{R/E} w \quad \text{et} \quad v \xrightarrow{}_{R,E} w$$

alors R est E-compatible.

Ce que l'on peut exprimer par le diagramme suivant:

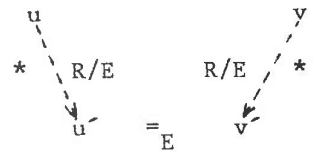
Les seules théories pour lesquelles on savait toujours assurer les conditions du lemme précédent étaient, jusqu'à maintenant, des théories dans lesquelles E exprimait l'associativité et la commutativité pour un nombre fini de symboles de fonctions. Nous étudierons dans la suite de cette thèse une autre théorie pour laquelle ces conditions sont également verifiées.

Pour énoncer une condition suffisante de E-confluence, nous avons encore besoin de la définition suivante:

DEFINITION 1.50: Un système de réécriture R sera dit E-clos si et seulement si il possède un ensemble complet de paires E-critiques PC tel que pour toute paire (u,v) dans PC, il existe u' et v' tels que

$$u \xrightarrow{*}_{R/E} u' \text{ et } v \xrightarrow{*}_{R/E} v' \text{ et } u' =_E v'.$$

Ce qui se visualise par le diagramme suivant:



Le théorème de Peterson et Stickel peut alors s'énoncer ainsi:

THEOREME 1.6: Si un système de réécriture R est à la fois E-noethérien et E-compatible, alors il est E-confluent si et seulement si il est E-clos.

Ce résultat est la base d'un algorithme de complétion qui généralise celui de Knuth et Bendix à des systèmes de réécriture E-canoniques. Sa description dans le cas d'axiomes commutatifs et associatifs est donnée dans [HUL,80].

Nous venons de rappeler deux des principales méthodes permettant d'étendre les résultats de Knuth et Bendix. Dans la suite de notre thèse, nous n'utiliserons que l'extension due à G.E.Peterson et M.E.Stickel, car la théorie que nous étudierons n'a pas la propriété de linéarité à gauche du système de réécriture de termes R. Il nous a parut néanmoins intéressant de donner un bref rappel des différentes techniques utilisables dans le contexte de théories auxquelles ne s'appliquent pas les résultats classiques.

CHAPITRE DEUX

LES ARBRES SIGNES

INTRODUCTION

A partir d'une signature donnée, nous construisons une nouvelle signature F obtenue en rajoutant un certain nombre de symboles de fonctions, parmi lesquels le symbole "-". L'ensemble des arbres signés AS est la F-algèbre libre engendrée par un ensemble de variables V. On se donne sur AS un ensemble d'axiomes A traduisant les propriétés de ces symboles de fonctions. Une algèbre signée est un élément de la variété des F-algèbres définies équationnellement par A. Le problème de la validité d'une équation dans une algèbre signée se ramène donc à l'étude de la théorie équationnelle $=_A$ dans la F-algèbre libre M(F,V).

Nous faisons cette étude en utilisant l'algorithme de complétion de Knuth et Bendix, que nous avons présenté dans le chapitre un. Cette théorie équationnelle est un exemple dans lequel l'algorithme de Knuth et Bendix ne termine pas, les axiomes générant, par complétion, un ensemble infini de règles de réécriture. Néanmoins, nous formalisons un concept nouveau, celui de méta-règle, chacune schématisant un ensemble infini de règles. L'ensemble des méta-règles nous permet alors de décrire finiment le système infini de règles de réécriture, et de prouver sa locale confluence.

Nous étudions successivement dans ce chapitre

* l'ensemble A des axiomes et le système de réécriture ER associé, en explicitant les choix qui ont été faits.

* la complétion du système de réécriture ER, en montrant que l'algorithme de complétion de Knuth et Bendix ne termine pas et engendre une infinité de règles.

* la formalisation du concept de méta-règle permettant de décrire l'ensemble infini de règles théoriquement engendré par complétion.

* un système de réécriture ECR ayant une infinité de règles, en prouvant que toute règle de ECR provient d'une paire critique générée par l'algorithme de Knuth et Bendix à partir de ER.

* la terminaison finie et la confluence de ECR, en décrivant pour terminer un algorithme performant de mise en forme normale d'un terme quelconque.

2.1- L'ENSEMBLE DES ARBRES SIGNES

DEFINITION 2.1: L'ensemble des arbres signés AS est la F-algèbre libre M(F,V) engendrée par V où

* V est un ensemble dénombrable de symboles de variables.

* pour tout entier i positif ou nul, F_i est un ensemble fini de symboles de fonctions d'arité i.

* F'_1 est un ensemble fini de symboles de fonctions unaires dont chaque élément noté f# est tel que f appartient à F_1.

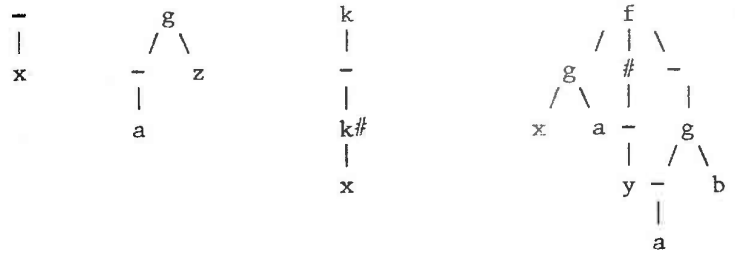
* "-" est un symbole de fonction unaire.

* F est la réunion finie de tous les F_i, de F'_1 et de {-}.

Les lettres w, x, y, z désignent dans la suite des variables, tandis que les symboles de fonctions 0-aires, encore appelés constantes, sont désignés par les lettres a, b, c, d; les lettres f, g, h, k sont réservées aux autres symboles de fonctions. Enfin s, t, u, v sont toujours des arbres signés.

Nous utiliserons indifféremment les notations -t et $\overset{|}{-}_{t}$.

EXEMPLES:



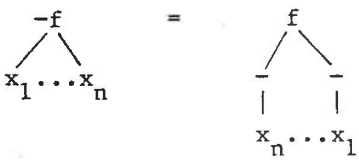
sont des arbres signés.

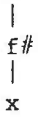
Informellement, le nouveau symbole - est introduit pour construire l'"opposé" -t d'un terme t et les symboles de F' peuvent être vus comme les inverses de chaque symbole de F_1.

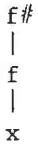
2.2- LES AXIOMES ET LE SYSTEME DE REECRITURE ASSOCIE

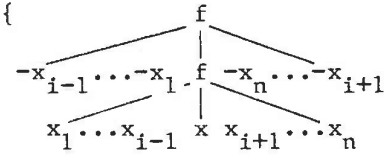
DEFINITION 2.2: On note A l'ensemble des axiomes suivants, définis sur AS:

(A1) = { $--x = x$ }

(A2) = {  pour tout f d'arité n>0 }

(A3) = {  pour tout f d'arité 1 }

(A'3) = {  pour tout f d'arité 1 }

(A4) = { 

pour tout f d'arité n>1 et pour tout i compris entre 1 et n }

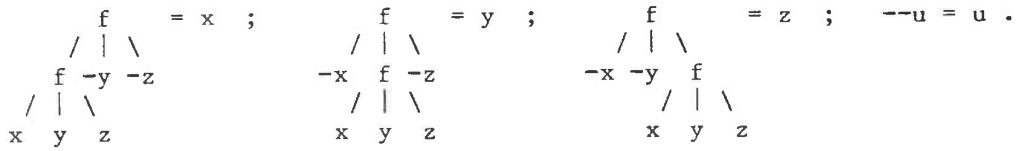
Intuitivement les axiomes de (A4) vont permettre de "simplifier" des termes et d'"isoler" le ième fils d'un terme en le composant avec les "opposés" des autres fils.

Les axiomes (A3) et (A'3) traduisent le fait que chaque symbole unaire est inversible et permettent également d'isoler son fils.

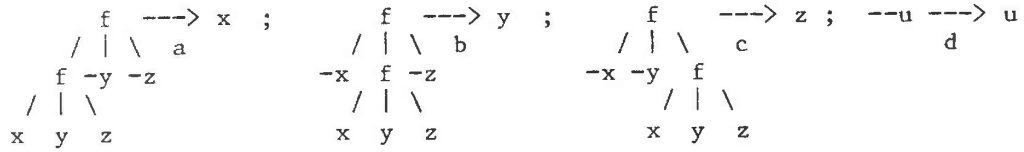
L'axiome (A1) peut s'interpréter par le fait que le symbole - est son propre inverse; associé aux axiomes de (A2), il va permettre de ne pas garder de signe - à un noeud qui n'est pas une feuille.

Faisons tout d'abord une remarque sur le choix de ces axiomes:

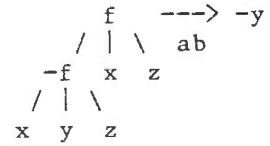
REMARQUE 2.1: supposons que l'on se soit donné les axiomes suivants:



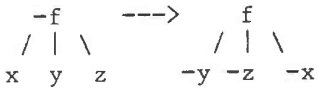
On en déduit les règles:



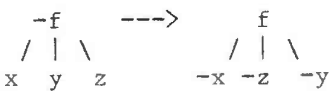
En cherchant les paires critiques, on trouve en superposant a dans b à l'occurrence 2 la règle ab:



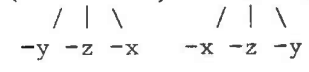
Puis en superposant ab dans a à l'occurrence 1 on obtient:



De même, on obtient en superposant c dans b à l'occurrence 2, puis la règle résultante dans c à l'occurrence 3, la règle:



On obtient donc la paire critique (f , f), ce qui

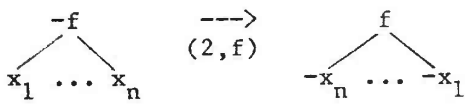


revient à ajouter des axiomes permutatifs. L'étude de telles théories équationelles est délicate et les résultats connus ne sont pas assez généraux pour être appliqués ici [HUE,81].

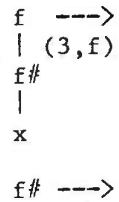
REMARQUE 2.2: On peut prouver, dans le cas où l'arité des symboles de fonctions est au plus trois, que les deux premiers axiomes sont une conséquence des axiomes de (A4). C'est certainement vrai dans le cas général, mais la preuve est techniquement très complexe: nous n'avons pas pu la faire, meme à l'aide du systeme FORMEL qui, déjà dans le cas de symboles d'arité trois, génère plus de deux cent règles.

Nous orientons les axiomes précédents de gauche à droite, pour obtenir le système de réécriture ER suivant:

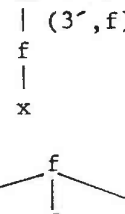
$$--x \longrightarrow x \quad (1)$$



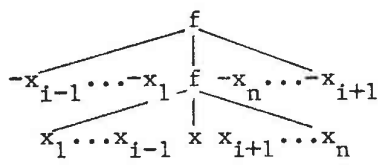
pour f d'arité supérieure ou égale à 1



pour f d'arité 1



pour f d'arité 1



où n>1 est l'arité de f et pour tout i = 1,...,n.

NOTATIONS: Nous noterons E l'ensemble des axiomes (A1) U (A2) et désignerons aussi par E le système de réécriture composé des ensembles de règles (1) et (2,f) pour chaque f appartenant à F.

Nous allons tout d'abord prouver que E est un système de réécriture canonique.

2.3- ETUDE DU SYSTEME DE REECRITURE E

Afin de prouver la confluence de E, nous étudierons successivement sa terminaison finie, puis sa locale confluence.

LEMME 2.1: E est à terminaison finie.

Preuve: Nous utilisons la méthode due à [DER,79] et rappelée dans le chapitre un. On définit l'ordre partiel suivant:

pour tout f appartenant à F, $f < -$.

Il induit un "ordre récursif sur les chemins" sur l'ensemble des termes, qui est un ordre de simplification noté $\overset{*}{<}$. L'ordre sur les multi-ensembles déduit de $\overset{*}{<}$ est noté $\overset{**}{<<}$. On va alors vérifier que, pour toute règle (g--->d), on a $d \overset{*}{<} g$.

En effet:

1) $x \overset{*}{<} --x$ car x est plongé dans --x .

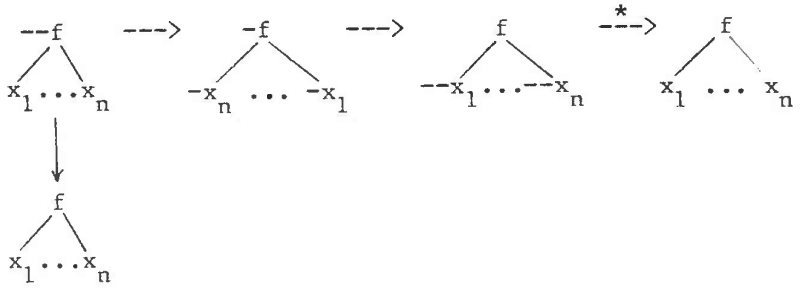
2) $f(-x_1,\ldots,-x_n) \overset{*}{<} -f(x_n,\ldots,x_1)$, car $- > f$

et $\{-x_1,\ldots,-x_n\} \overset{**}{<<} \{ -f(x_1,\ldots,x_n) \}$ puisque

$-x_i$ est plongé dans $-f(x_1,\ldots,x_n)$ pour tout i=1,...,n. []

LEMME 2.2: E est localement confluent.

Preuve: Il suffit de calculer les paires critiques, et le seul cas possible est de superposer une règle de (2) sur la règle de (1) à l'occurrence 1. Nous utilisons une notation arborescente pour une meilleure lisibilité.

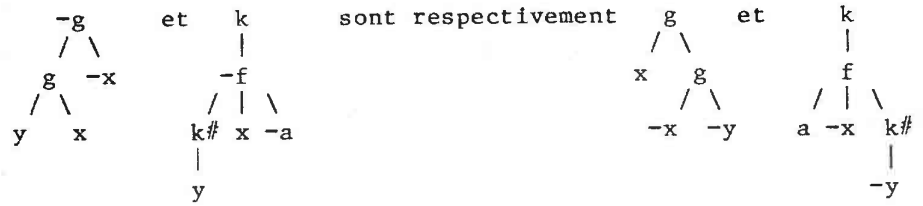


et cette superposition ne génère pas de nouvelle règle. []

COROLLAIRE 2.1: E est confluent.

DEFINITION 2.3: Un terme t est dit en pré-forme normale s'il est en forme normale pour les règles de E. On notera pfn(t) la pré-forme normale du terme t.

EXEMPLE : Les pré-formes normales des termes

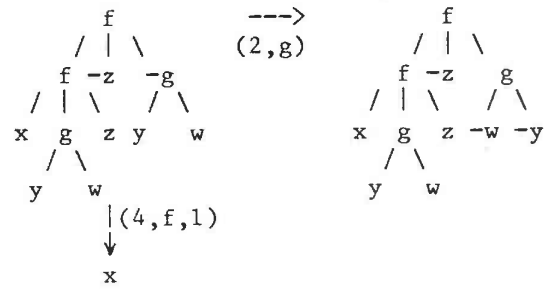


2.4- COMPLETION DU SYSTEME DE REECRITURE ER ASSOCIE AUX AXIOMES

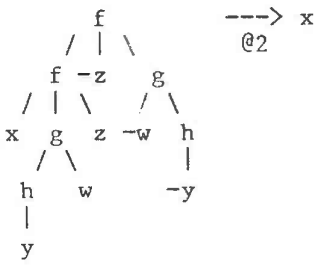
Il est facile de constater que le système de réécriture ER n'est pas confluent, comme le prouve l'exemple suivant.

Soient f, g, h trois symboles de fonctions d'arités respectives 3, 2, 1 et les règles correspondantes.

La superposition de la règle (2,g) sur la règle (4,f,1) à l'occurrence 3 donne:



En orientant par plongement la paire critique obtenue,on obtient la règle @1, qui est rajoutée au système de réécriture; on peut alors superposer la règle (2,h) sur le membre gauche de @1 à l'occurrence 32 et en déduire la nouvelle règle @2:



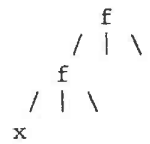
Un tel processus peut se poursuivre indéfiniment à cause des règles (2,f). Un ensemble infini de règles d'une complexité croissante est alors engendré, pour chaque symbole de fonction d'arité supérieure à 1.

Il est légitime de se demander alors si une autre orientation des axiomes peut conduire à une situation différente. Seul le second axiome peut être orienté autrement sans enlever de facon évidente au système de réécriture la propriété de terminaison finie. Mais le système obtenu en remplacant chaque règle (2,f) par la règle

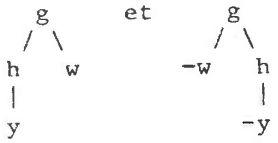
$$f(-x_n,\ldots,-x_1) \longrightarrow -f(x_1,\ldots,x_n)$$

n'est pas complet et l'algorithme de complétion de Knuth et Bendix engendre encore une fois une infinité de règles. Dans la première orientation, que nous avons choisie, un terme en forme normale a tous ses symboles - portés par ses feuilles.

Au vu des règles engendrées, on constate que l'on obtient ainsi des familles de règles ayant dans leur membre gauche le même squelette, par exemple dans le cas ci-dessus



et dont les autres sous-termes présentent une certaine symétrie, comme c'est le cas pour -z et z



C'est cette constatation que nous allons formaliser dans la suite.

2.5- INTRODUCTION DES META-REGLES

Intuitivement, une méta-règle est un schéma de règles permettant de décrire une famille infinie de règles. Plus précisément, le membre gauche d'une méta-règle est fabriqué à l'aide d'un squelette, commun à cette famille infinie de règles, et de sous-termes ayant des liens entre eux; ces liens seront matérialisés par un symbole fonctionnel, ici noté m. Lorsqu'on voudra passer de la méta-règle à une règle, il suffira d'instancier certaines variables par des termes appelés structures, puis de calculer, à l'aide de l'opérateur m, et ce jusqu'à ce qu'il disparaisse; m devra donc vérifier un principe de définition.

C'est cette idée que nous allons préciser dans ce paragraphe.

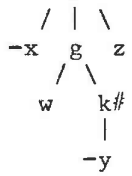
2.5.1- LES STRUCTURES DE AS

Tout d'abord nous définissons un sous-ensemble particulier de termes de AS, appelés structures, qui sont des arbres dont toutes les feuilles sont des variables distinctes et où tous les signes "-" sont portés par des feuilles; de facon plus précise:

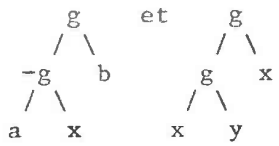
DEFINITION 2.4: L'ensemble ST des structures est défini inductivement par:

* V est inclus dans ST
* si x appartient à V, alors -x appartient à ST
* si $s_1, \ldots, s_n$ sont des structures telles que pour tout couple d'entiers distincts i et j compris entre 1 et n, $V(s_i)$ et $V(s_j)$ sont disjoints, alors $f(s_1,\ldots,s_n)$ est une structure, quel que soit f d'arité n.

EXEMPLES :



est une structure.



ne sont pas des structres.

Remarquons qu´une structure est un terme en pré-forme normale.

2.5.2- LE PRINCIPE DE DEFINITION

Nous proposons ici un principe de définition analogue à celui énoncé par Huet et Hullot [H&H,80] dans le cadre de l´algèbre initiale, et que nous appliquons à des F-algèbres libres.

Soit F un ensemble de symboles de fonctions avec arités, M(F,V) la F-algèbre libre engendrée par un ensemble dénombrable de variables V, H un ensemble fini de symboles de fonctions et M(FUH,V) la FUH-algèbre libre engendrée par V.

PRINCIPE DE DEFINITION:
Soit M un ensemble d´équations sur M(FUH,V) et $=_M$ la congruence engendrée. On dit que M définit H sur F si et seulement si, pour tout t appartenant à M(FUH,V), il existe t´ unique appartenant à M(F,V) tel que $t =_M t'$.

Il est facile de voir que cette condition est équivalente aux suivantes:

* pour tout t dans M(FUH,V), il existe t´ dans M(F,V) tel que $t =_M t'$.
* pour tous t et t´ de M(F,V), $t =_M t'$ implique $t = t'$.

2.5.3- DEFINITION DE LA FONCTION m

Nous allons définir un nouvel opérateur dans AS, note m, et dont la signification intuitive est d´exprimer la symétrie de certains sous-termes dans les règles engendrées par l´algorithme de complétion.

Soit m un symbole de fonction unaire et AS^e la FU{m}-algèbre libre engendrée par V.

Désignons par M l'ensemble des équations suivantes dans AS^e :

* pour tout x appartenant à F_0UV, $m(x) = -x$ et $m(-x) = x$

* pour tout f d'arité n, distinct de - et de m,

$$m(f(x_1,\ldots,x_n)) = f(m(x_n),\ldots,m(x_1))$$

$$m(-(f(x_1,\ldots,x_n))) = f(m(-x_1),\ldots,m(-x_n))$$

* $m(-(-x)) = m(x)$.

PROPOSITION 2.1: M définit m sur F.

Preuve: soit R_M le système de réécriture obtenu à partir de M en orientant les axiomes de gauche à droite.

* R_M termine car une dérivation issue d'un terme t a une longueur bornée par la taille de t.

* On prouve facilement, par récurrence structurelle, que la forme normale d'un terme de AS^e est unique pour le système de réécriture R_M.

Pour chaque terme t, nous noterons t'= M-FN(t) son unique R_M-forme normale. Il est clair que t' appartient à AS, puisque le symbole m n'apparait plus dans t'.

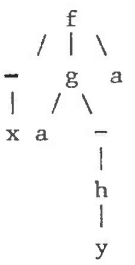
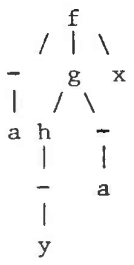
Enfin, puisque chaque membre gauche de règle dans R_M a pour symbole de tête m, un terme de AS est forcément irréductible et égal à sa M-forme normale. []

On définit alors sur AS l'opérateur m_*.

DEFINITION 2.5: L'opérateur m_* de AS dans AS associe à un terme t le terme $m_*(t)$ défini par

$$m_*(t) = M\text{-}FN(m(t))$$

Muni de cette nouvelle opération, AS est une FU{m}-algèbre. Par la suite, nous noterons encore m l'opérateur m_* dans AS et nous dirons que m(t) est le terme miroir du terme t.

EXEMPLE : si t est le terme  , m(t) est le terme 

REMARQUE: Il est facile de constater, et nous le prouvons dans le paragraphe suivant, que m calcule la pré-forme normale de -t. Il aurait été possible de définir directement m ainsi dans AS. Mais nous avons préféré utiliser ce formalisme, en vue de pouvoir ensuite proposer une définition générale d'une méta-règle.

2.5.4- PROPRIETES DE L'OPERATEUR m DANS AS

La définition de m que nous avons donnée est peu utilisable dans les preuves et nous allons caractériser autrement m(t), en précisant le lien existant entre l'opérateur m et les axiomes de E.

LEMME 2.3: Soit t un terme de AS; alors $-t \xrightarrow{*}{}^{E} m(t)$

Preuve: par récurrence structurelle sur t:

* Si $t \in F_0 \cup V$, c'est clair.

* Si $t = f(t_1,\ldots,t_n)$, $-t \rightarrow^{E} f(-t_n,\ldots,-t_1)$ qui se réécrit en utilisant l'hypothèse de récurrence en $f(m(t_n)\ldots m(t_1))$ lui même égal, par définition de m, à m(t).

* Si t=-u, alors soit u=-v et $-t=---v \rightarrow^{E} -v$ qui se réécrit par hypothèse de récurrence en m(v)=m(t).

soit $u= f(u_1,\ldots,u_n)$ et -t se réécrit en $f(--u_1,\ldots,--u_n)$, puis, par hypothèse de récurrence, chaque sous-terme $--u_i$ se réécrit en $m(-u_i)$. Donc -t se réécrit en $f(m(-u_1),\ldots,m(-u_n)) = m(t)$. []

PROPOSITION 2.2: Pour tout terme t appartenant à AS, $m(t) = pfn(-t)$

Preuve: elle résulte du fait que m(t) est irréductible pour les règles de E et de l'unicité de la E-forme normale de t. []

Une autre propriété importante de l'opérateur m est qu'il est idempotent sur les termes en pré-forme normale; cette propriété est utile dans les preuves de confluence que nous ferons ensuite.

LEMME 2.4: Pour tout terme t appartenant à AS, $t \xrightarrow{*}_E m(m(t))$.

Preuve: $m(m(t)) = pfn(-(pfn(-t))) = pfn(--t) = pfn(t)$. []

PROPOSITION 2.3: Si t est un terme de AS en pré-forme normale, $t = m(m(t))$.

Preuve: elle est immédiate à partir du lemme précédent et du fait que t et m(m(t)) sont tous les deux en pré-forme normale. []

Enfin nous avons besoin, pour la suite, de préciser le comportement de l'opérateur m par rapport aux substitutions.

LEMME 2.5: Soient σ une substitution dont l'image de chaque variable est en pré-forme normale et t un terme;
si $\sigma(t) \xrightarrow{*}_E t'$, t' en pré-forme normale, alors $\sigma(m(t)) \xrightarrow{*}_E m(t')$.
ce qui s'exprime encore par:
si $t'=pfn(\sigma(t)$ alors $m(t')=pfn(\sigma(m(t))$

Preuve: par récurrence structurelle sur t:

* t=y ou -y, y∈V et $y \notin D(\sigma)$, alors la conclusion est évidente.

* t=x, x∈V et $x \in D(\sigma)$: $\sigma(x) \xrightarrow{*}_E t'$ d'où
$\sigma(-x)=-\sigma(x) \xrightarrow{*}_E -t' \xrightarrow{*}_E m(t')$ d'après le lemme 2.3.

* t=-x, x∈V et $x \in D(\sigma)$ avec $\sigma(x)=u$: $\sigma(t)=\sigma(-x)=-u \xrightarrow{*}_E m(u)$
par application du lemme 2.3. D'autre part $\sigma(m(t))=\sigma(x)=u=m(m(u))$
car la fonction m est idempotente sur u en pré-forme normale.

* $t = f(t_1,\ldots,t_n)$, $\sigma(t) = f(\sigma(t_1),\ldots,\sigma(t_n)) \xrightarrow{*}{}^{E} t'$;

donc t' est de la forme

$f(t'_1,\ldots,t'_n)$ où pour $i=1,\ldots,n$, $\sigma(t_i) \xrightarrow{*}{}^{E} t'_i$

Alors $\sigma(m(t)) = \sigma(f(m(t_n),\ldots,m(t_1)) = f(\sigma(m(t_n)),\ldots,\sigma(m(t_1)))$

se réécrit en $f(m(t'_1),\ldots,m(t'_n))$ par hypothèse de récurrence.

D'où $(m(t)) \xrightarrow{*}{}^{E} m(f(t'_n,\ldots,t'_1) = m(t')$.

* $t=-f(t_1,\ldots,t_n)$, $\sigma(t) = -f(\sigma(t_1),\ldots,\sigma(t_n)) \xrightarrow{*}{}^{E} t'$.

Donc t' est de la forme $f(t'_n,\ldots,t'_1)$, avec

$\sigma(-t_i) \xrightarrow{*}{}^{E} t'_i$ pour $i=1,\ldots,n$.

Alors $\sigma(m(t)) = f(\sigma(m(-t_1)),\ldots,\sigma(m(-t_n)))$ se réécrit, par

hypothèse de récurrence, en $f(m(t'_1),\ldots,m(t'_n)) = m(t')$. []

2.5.5- DEFINITION DES META-REGLES DANS AS

Nous proposons tout d'abord une définition générale d'une méta-règle que nous particulariserons ensuite aux arbres signés.

DEFINITION 2.6: Une méta-règle est définie par la donnée

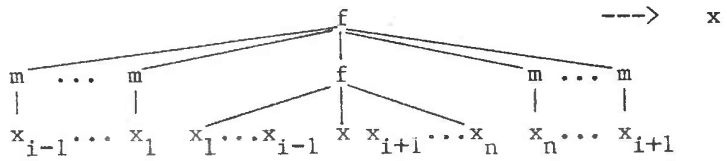
* d'un ensemble H de symboles vérifiant le principe de définition

* d'une règle G--->D dans M(FUH,V)

* d'un sous-ensemble de ses variables, dont l'instanciation par des structures permet de calculer les règles sur M(F,V) associées.

Pour simplifier, on appellera méta-règle la règle G--->D.

Dans AS, considérons l'ensemble des méta-règles suivantes:



pour chaque f de F d'arité n strictement supérieure à 1 et pour chaque i compris entre 1 et n.

La méta-règle précédente sera appelée méta-règle [4,f,i].
Nous lui associons le sous-ensemble W de ses variables qui figurent sous une occurrence quelconque du symbole m. Ici, $W = \{x_1,\ldots,x_{i-1},x_{i+1},\ldots x_n\}$.

Remarquons que, puisque m vérifie le principe de définition, nous venons bien de définir des méta-règles sur AS.

Il est alors possible de proposer une relation de "méta-réécriture" en précisant ce que l'on entend par appliquer une méta-règle sur un terme de AS:

DEFINITION 2.7: Un terme t en pré-forme normale se réécrit à l'occurrence p avec la méta-règle G--->D en un terme t' si et seulement si
il existe une substitution σ telle que

$$t_{|p} = \text{M-FN}(\sigma(G))$$
$$\text{et} \quad t' = t[p \leftarrow \text{M-FN}(\sigma(D))].$$

On notera alors $t \xrightarrow{M} t'$.

REMARQUE: Sans imposer à t d'être en pré-forme normale, on peut donner une définition analogue en posant:

$$t_{|p} =_E \text{M-FN}(\sigma(G)) \quad \text{et} \quad t' = t[p \leftarrow \text{M-FN}(\sigma(D))].$$

Cette définition ne donne pas une procédure de décision de la méta-réductibilité. Il faut donc la compléter par un autre résultat:

PROPOSITION 2.4: Dans AS, la méta-réductibilité est décidable.

Preuve: une procédure de décision pour l'application de la méta-règle [4,f,i] à l'occurrence p dans le terme t est fournie par le procédé suivant:

* on calcule la pré-forme normale de t, notée t'.
* on cherche un filtre σ de $G_{|i}$ vers $t'_{|p.i}$
* s'il existe, on vérifie que

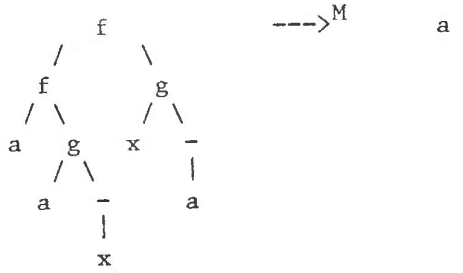
pour tout entier j compris entre 1 et i-1,

$$t'_{|p.j} = m(t'_{|p.i.(i-j)}) \text{ et}$$

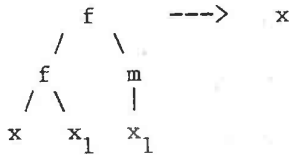
pour tout entier j compris entre i+1 et n,

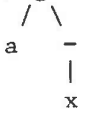
$$t'|_{p.j} = m(t'|_{p.i.(n+i+1-j)}).$$ []

EXEMPLE :



en utilisant à l'occurrence ε la méta-règle



avec la substitution (x-->a) (x_1 --> ).

2.5.6- ENSEMBLE DE REGLES ASSOCIE A UNE META-REGLE

Nous allons maintenant préciser comment une méta-règle décrit une infinité de règles. Pour cela nous utiliserons un ensemble d'applications de V dans l'ensemble ST des structures. Une telle application se prolonge de facon unique en un endomorphisme de AS^e.

DEFINITION 2.8: A la méta-règle G--->D et à l'ensemble W de variables correspondant de la définition 2.6, on associe l'ensemble de toutes les règles

M-FN(α (G)) ---> D telle que α soit une application de W dans ST .

Nous noterons [IV,f,i] l'ensemble de règles correspondant à la méta-règle [4,f,i].

REMARQUES: On aurait pu écrire l'ensemble des règles associées à la méta-règle G--->D sous la forme M-FN(α (G)) ---> M-FN(α (D)) puisque, avec la définition de α choisie, M-FN(α (D)) = D.

D'autre part, il faut noter que l'ensemble [IV,f,i] est dénombrable, puisque l'ensemble des applications α l'est.

Pour relier toutes ces notions entre elles, il faut établir le résultat suivant:

PROPOSITION 2.5: Soit t un terme de AS en pré-forme normale. La méta-règle [4,f,i] s'applique à l'occurrence p dans le terme t pour obtenir le terme t' si et seulement si il existe une règle de l'ensemble [IV,f,i] qui réduit t en t' à l'occurrence p.

Preuve: pour plus de clarté, nous supposerons que $p=\varepsilon$.

* si $t \rightarrow t'$, il existe des substitutions σ' et α telles que

$t' = \sigma'(M\text{-}FN(\alpha(G)))$

$= M\text{-}FN(\sigma'\alpha(G))$ et il suffit de prendre $\sigma = \sigma'\alpha$ pour obtenir:

$t \rightarrow^{M} \sigma(D) = \sigma'\alpha(D) = \sigma'(D) = t'$.

* si $t \rightarrow^{M} t'$, il existe σ telle que $\sigma(G_{|i}) = t_{|i}$.

donc pour chaque variable x_j appartenant à W, il existe une structure s_j de ST et une substitution σ' à valeurs dans F_0UV, telles que $\sigma'(s_j) = \sigma(x_j)$. On pose alors

pour tout $j \neq i$ $\alpha(x_j) = s_j$, puis $\sigma'(x_i) = \sigma(x_i)$

et il est facile de vérifier ensuite que

$t_{|i} = \sigma'\alpha(x_j)$ et

$t_{|j} = \sigma'(M\text{-}FN(\alpha(G_{|j})))$ pour tout $j = 1,\ldots,i-1,i+1,\ldots n$.

Donc $t = \sigma'(M\text{-}FN(\ (G))$ et $t \rightarrow \sigma'(D) = t'$. []

Cette dualité permet donc de travailler avec l'une ou l'autre des deux réécritures. Mais prouver la confluence du système avec méta-règles nécessite de développer des outils appropriés, par exemple une généralisation de la méthode de Knuth et Bendix à des méta-règles. Ce ne sera pas notre but dans cette thèse et nous n'utiliserons les méta-règles qu'en temps que description finie d'ensembles infinis de règles, ce qui permettra en outre d'alléger certaines preuves.

2.6- LE SYSTEME DE REECRITURE CANONIQUE

Dans tout ce paragraphe, nous utiliserons fréquemment la notation arborescente pour assurer une meilleure lisibilité.

DEFINITION 2.9: Nous noterons ECR l'ensemble des règles de réécriture definies sur AS et réunion des ensembles suivants:

(1) = { --x ---> x }

(2) = { $-f(x_1 \dots x_n)$ ---> $f(-x_n \dots -x_1)$ | $f \in F \setminus F_0$ }

(3) = { $f(x) \rightarrow f\#(x)$ | $f \in F_1$ }

(3') = { $f\#(x) \rightarrow f(x)$ | $f \in F_1$ }

(4) = { [IV,f,i] | $f \in F_n$, $n>1$, $i=1,\dots,n$ }

(1), (2), (3) et (3') sont des ensembles finis de règles mais (4) étant un ensemble infini (dénombrable), nous avons donc ici un système infini de règles de réécriture. Aussi devons nous nous assurer, avant de poursuivre l'étude de ECR, que:

LEMME 2.6: Il est possible de décider si un terme est en forme normale pour ECR.

> Preuve: Un terme t étant donné, le nombre de règles dont le membre de gauche est plus petit ou égal à la taille de t est fini puisque le nombre de symboles d'arité non nulle est fini. Le nombre de règles applicables à t est donc fini. []

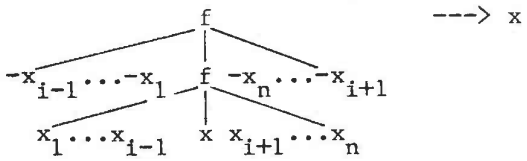
REMARQUE: Avec la notion de méta-réécriture, ce résultat devient évident et découle de la définition 2.8 et de la proposition 2.5.

Montrons tout d'abord que ce système ne contient que des règles théoriquement engendrées par l'algorithme de complétion à partir des axiomes.

LEMME 2.7: Chaque règle de (4) vient d'une paire critique générée par l'algorithme de Knuth et Bendix appliqué à ER.

Preuve: par récurrence sur la somme p des tailles des structures substituées aux variables des méta-règles, pour f fixé.

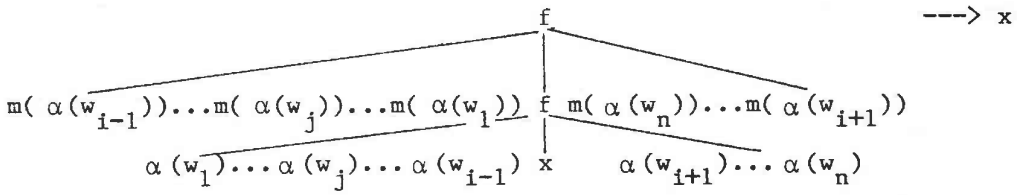
* p=0. Il s'agit de la règle



et c'est l'une des règles initiales.

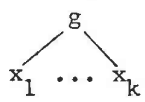
* Supposons le résultat démontré jusqu'à p et montrons le pour p+1.

Considérons une règle de [IV,f,i] dont le membre de gauche est donc de taille 2*(p+1)+2.



Alors il existe $\alpha(w_j)$ ayant au moins un sous terme u de taille 1.

u est de la forme



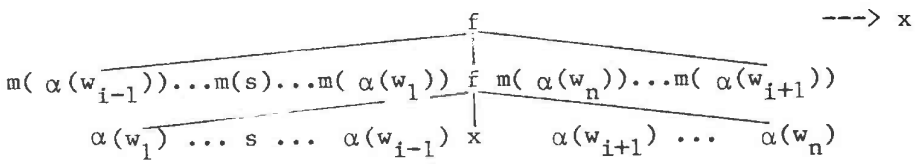
avec g appartenant à F, ou de la forme -x, x étant dans V.

Remplacons u dans $\alpha(w_j)$ par z, z n'étant pas une variable de $\alpha(w_j)$: on obtient ainsi une structure de taille p notée s.

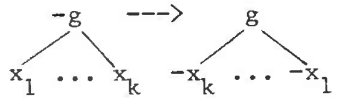
Comme le miroir de $\alpha(w_j)$ apparait aussi dans le membre gauche de la règle de [IV,f,i] considérée, remplacons m(u) par -z;

Si σ'' est la substitution (z-->u), on a alors: $\sigma''(s) = \alpha(w_j)$ et $\sigma''(m(s)) \xrightarrow{*}{}^{E} m(\alpha(w_j))$ par le lemme 2.5.

Considérons donc la règle



qui par hypothèse de récurrence provient d'une paire critique engendrée par l'algorithme de Knuth et Bendix. Comme -z appartient au membre gauche de cette règle, on peut superposer la règle



ou la règle --x ---> x

sur -z pour obtenir, après application du lemme 2.3, les membres gauche et droit de la règle considérée au départ.

Ce qui termine la preuve de ce lemme. []

Nous nous proposons maintenant d'étudier la confluence de ECR.

LEMME 2.8: ECR est à terminaison finie.

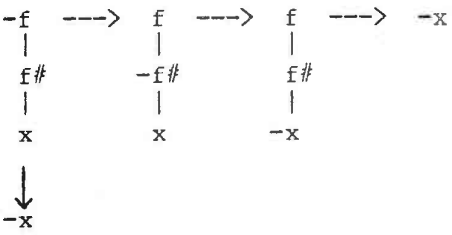
Preuve: La méthode et l'ordre sont les mêmes que précédemment. Les règles de (3), (3') et (4) vérifient trivialement $d \overset{*}{<} g$ puisque d est sous-terme de g. []

La confluence de E étant prouvée, il reste à superposer les règles de E sur celles de (3)U(3')U(4) et ces dernières entre elles, afin de prouver par le théorème de Knuth et Bendix que ECR est localement confluent.

LEMME 2.9: La superposition des règles de E avec celles de (3)U(3')U(4) ne génère pas de nouvelles règles.

Preuve: les seules superpositions possibles sont

* la superposition d'une règle de (3) dans une règle de (2) ayant même symbole de tête, à l'occurrence ε.



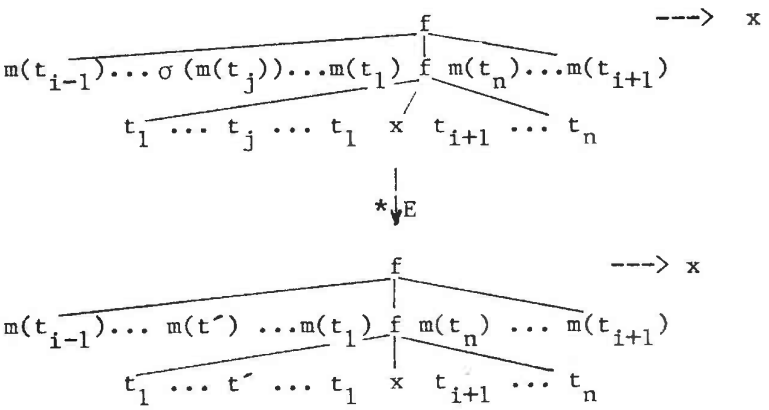
et la paire critique est trivialement confluente.

* la superposition d'une règle de (3') avec une règle de (2) qui se traite de facon similaire.

* la superposition de la règle (1) dans une règle de (4).

Soit σ l'unificateur principal du sous-terme $-w_j$ dans la structure t_j avec $--x$.

$\sigma(t_j) \xrightarrow{*}_E t'$ en pré-forme normale, et $\sigma(m(t_j)) \xrightarrow{*}_E m(t')$ par le lemme 2.5. D'où

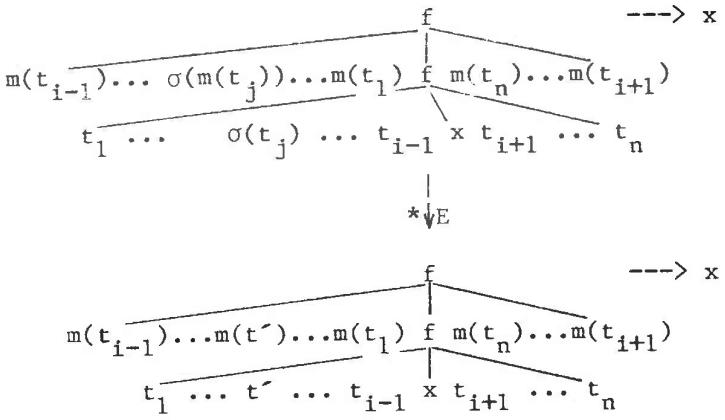


* la superposition d'une règle de (2) dans une règle de (4).

Soit σ l'unificateur principal de $-w_j$ dans une structure t_j avec $-g(x_1,\ldots,x_k)$ et en conséquence

$\sigma(t_j) \xrightarrow{*}_E t'$ en pré-forme normale

et $m(\sigma(t_j)) \xrightarrow{*}_E m(t')$ par le lemme 2.5. Donc



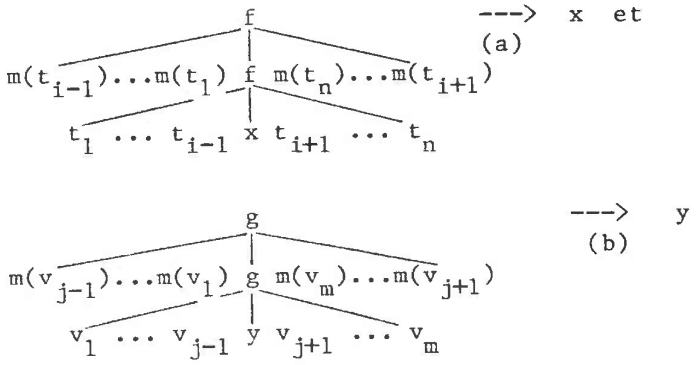
Aucune règle nouvelle n'est donc engendrée par ces superpositions. []

LEMME 2.10: La superposition de deux règles de (3)U(3')U(4) n'engendre pas de nouvelle règle.

Preuve: Il est facile de voir, en utilisant le lemme 2.5, que la superposition d'une règle de (3) ou de (3') dans une règle de (4) ne génère pas de nouvelle règle.

Il reste donc à superposer de toutes les facons possibles les règles de (4) entre elles.

Donnons nous les deux règles suivantes:

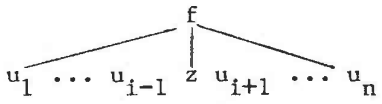


Comme conséquence du lemme 2.5, il est clair que, si on unifie v_k avec le membre gauche de (a), on n'engendre pas de nouvelle règle, quand f est différent de g.

Le seul cas restant est celui résultant de la superposition des règles de (4) ayant même symbole de fonction et trois possibilités sont alors

à envisager.

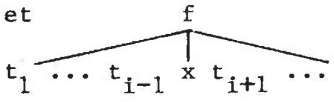
* Soit on superpose (a) sur elle-même à l'occurrence i, auquel cas l'unificateur σ de



avec le membre gauche t de (a) unifie

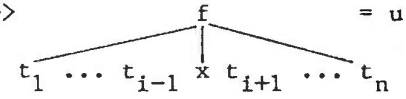
u_1 et $m(t_{i-1})$ ... u_{i-1} et $m(t_1)$

z et

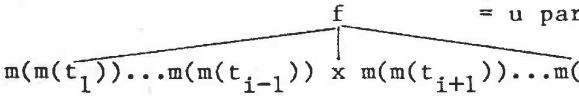


u_{i+1} et $m(t_n)$ ... u_n et $m(t_{i+1})$

D'où $\sigma(t)$ ---> (b)

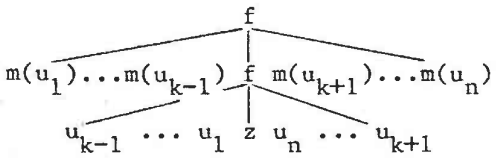


= u



= u par la proposition 2.3.

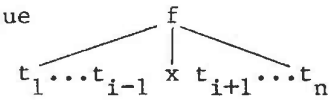
* Soit on superpose la règle



---> z (c)

à l'occurrence i dans la règle (a).

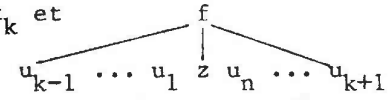
En supposant par exemple que k<i et que



s'unifie avec le membre gauche t' de la règle (c), l'unificateur σ' unifie

t_1 et $m(u_1)$ ... t_{k-1} et $m(u_{k-1})$

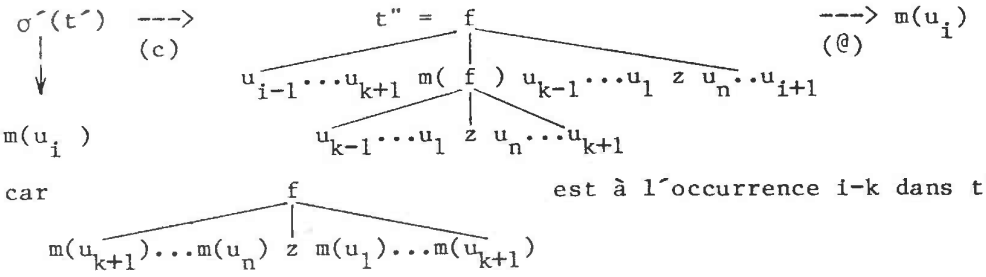
t_k et



t_{k+1} et $m(u_{k+1})$ ... t_{i-1} et $m(u_{i-1})$

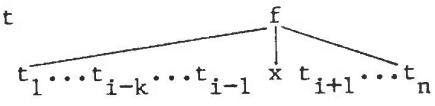
x et $m(u_i)$, t_{i+1} et $m(u_{i+1})$ ... t_n et $m(u_n)$.

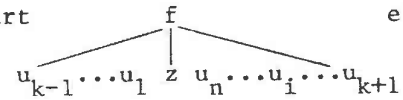
En tenant compte du fait que la fonction m est idempotente sur les structures, on obtient



car est à l´occurrence i-k dans t".

* la superposition de (a) dans (c) à l´occurrence ε est impossible, car l´unificateur des membres gauches de ces deux règles devrait alors unifier

d´une part $m(u_i)$ et 

d´autre part  et $m(t_{i-k})$

et il est aisé de se convaincre que cela est impossible.

Ce qui termine la preuve. []

Les lemmes qui précèdent permettent d´énoncer le résultat suivant:

COROLLAIRE 2.2: ECR est localement confluent.

PROPOSITION 2.6: ECR est confluent et à terminaison finie.

Sachant déjà qu´il est possible de décider si un terme est en forme normale, et sinon d´en calculer la forme normale par réécriture, nous décrivons maintenant un algorithme de calcul efficace. Il calcule la forme normale d´un terme déjà en pré-forme normale, et qui est donc
soit un atome signé (constante ou variable éventuellement précédée du signe -), soit de la forme $f(t_1,\ldots,t_n)$.

PROPOSITION 2.7: L'algorithme suivant calcule la forme normale de tout terme t en pré-forme normale.

FN(t) <-- SI t est un atome signé ALORS t

SINON $t = f(t_1,\ldots,t_n)$

t'_1 <- FN(t_1)

...

t'_n <- FN(t_n)

CAS . une règle de (3) ou (3') s'applique

ALORS fils de t'_1

. une règle de [IV,f,i] s'applique

ALORS i-ième fils de t'_i

. SINON $f(t'_1,\ldots,t'_n)$

FIN CAS

FIN SI

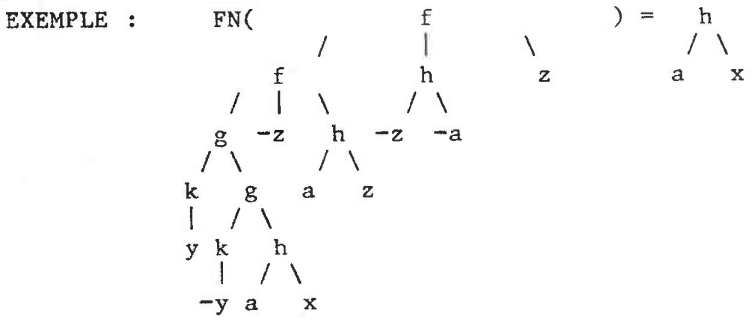
Preuve: par induction structurelle sur t.

* Si t est un atome signé, il est irréductible d'où la conclusion.

* Si $t = f(t_1,\ldots,t_n)$, alors $t \xrightarrow{*} t' = f(t'_1,\ldots,t'_n)$ avec $t'_i = FN(t_i)$ par hypothèse de récurrence. Il suffit alors de tester si une règle s'applique à l'occurrence ε.

C'est exactement ce que fait l'algorithme. []

EXEMPLE :



Disons quelques mots au sujet du coût de cet algorithme; nous allons prouver que, si n est la taille de l'arbre t, le coût c(n) est majoré par une fonction quadratique de n.

Désignons par N l'arité maximale des symboles de fonctions de t.

Le coût du calcul des formes normales de chaque fils est majoré par $c(n_i)$ où les n_i sont tels que $\Sigma_i n_i = n-1$.

D'autre part, le coût du test d'application d'une règle est , dans le pire cas où f est un symbole d'arité maximale N, majoré par N*n, n étant un majorant du nombre de tests faits pour vérifier que les sous-termes sont miroirs. (On pourrait en fait prendre n-2).

Nous obtenons donc, K étant une constante:

$$c() \leq \sum_{n_i \mid \Sigma_i n_i = n-1} c(n_i) + n*N + K$$

Prouvons par récurrence sur n que $c(n) \leq (N+K) * n^2$

* pour n=1, il est clair que l'algorithme est appelé au plus N fois et qu'aucune règle ne s'applique, donc c(1) = N+K.

* si l'on suppose que $c(n_i) \leq (N+K) * n_i^2$ pour chacun des fils de f,

$$c(n) \leq \sum_{n_i \mid \Sigma_i n_i = n-1} (N+K) * n_i^2 + n*N + K \leq (N+K) * (n-1)^2 + n*N + K$$

puisque $n_1^1+\ldots+n_p^2 \leq (n_1+\ldots+n_p)^2$ pour $n_1,\ldots,n_p \geq 0$.

Donc $c(n) \leq (N+K)*n^2 - N*(n-1) - 2*K*(n-1) \leq (N+K)*n^2$ pour n>1.

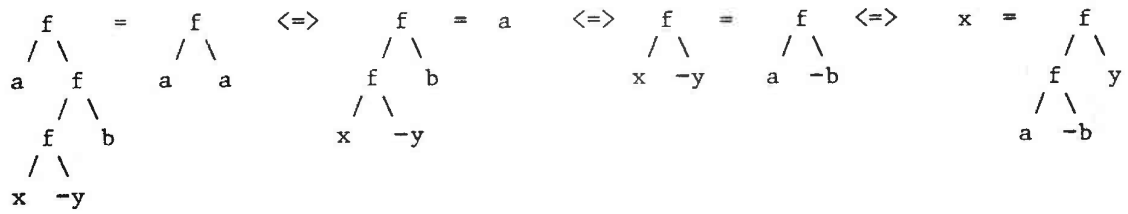
Dans le cas d'arbres binaires, il est possible d'estimer les coûts des tests de miroirs de facon plus précise que dans le cas général et de prouver que le coût de l'algorithme est proportionnel à nlog(n) [KKJ,81].

CHAPITRE TROIS

RESOLUTION D'EQUATIONS LINEAIRES DANS LES ARBRES SIGNES

INTRODUCTION

La théorie équationnelle définie sur les arbres signés a pour but de permettre des transformations de type algébrique sur les équations: en composant les deux membres de l'équation par le même opérateur et avec le même terme, on obtient une équation équivalente; par ce procédé, il est possible d'isoler n'importe quel sous-terme. Ce chapitre a pour but de justifier les suites d'équivalences du type suivant:



où la première étape est une simplification d'équation, et où les autres étapes permettent d'isoler la variable x; la dernière équation fournit la plus petite solution de l'équation de départ.

Nous allons successivement

* justifier les transformations d'équations, en montrant qu'elles conservent l'ensemble des unificateurs,

* définir une méthode de simplification d'équations, permettant de réduire la taille d'une équation sans changer l'ensemble de ses solutions,

* résoudre les équations linéaires et donc toutes les équations simplifiables en une équation linéaire,

* envisager le problème du filtrage qui revient aussi à résoudre une équation mais en ne s'autorisant à instancier que le premier membre.

3.1- EQUATIONS EQUIVALENTES

Dans ce paragraphe nous allons étudier des transformations conservant l'ensemble des solutions d'une équation, dans le but de nous donner des outils permettant dans la suite de simplifier des équations sans modifier l'ensemble des solutions.

Rappelons que deux équations $(t=t')$ et $(u=u')$ sont équivalentes si et seulement si elles ont le même ensemble de solutions.

Notons également que nous ne considérons pas comme équivalentes deux équations qui ne diffèrent que par un renommage de leurs variables.

LEMME 3.1: Si $t=_A u$ et $t'=_A u'$, alors les deux équations $(t=t')$ et $(u=u')$sont équivalentes.

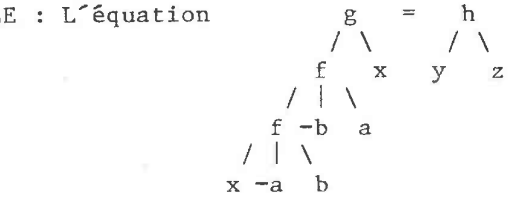
Preuve: si $t=_A u$ et $t'=_A u'$, alors pour toute substitution σ

$\sigma(t) =_A \sigma(t')$ si et seulement si $\sigma(u) =_A \sigma(u')$. []

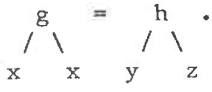
En conséquence,

COROLLAIRE 3.1: t et t' étant deux termes de AS l'équation $(t=t')$ est équivalente à l'équation $(FN(t)=FN(t'))$.

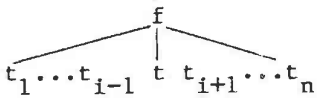
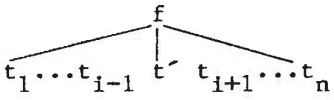
EXEMPLE : L'équation



est équivalente à



LEMME 3.2: Pour tout symbole f d'arité n non nulle, et pour tous termes $t_1,\ldots,t_n$, t, t':

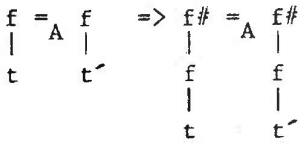
$t =_A t'$ <=>  $=_A$ 

pour tout i = 1,...,n.

Preuve: L'implication directe est claire, car $=_A$ est une congruence.

Prouvons la réciproque:

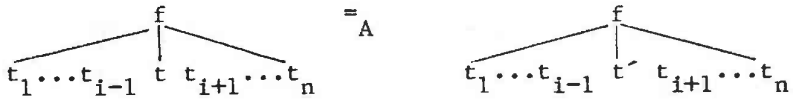
* Si f est un symbole d'arité 1, appartenant par exemple à F_1,



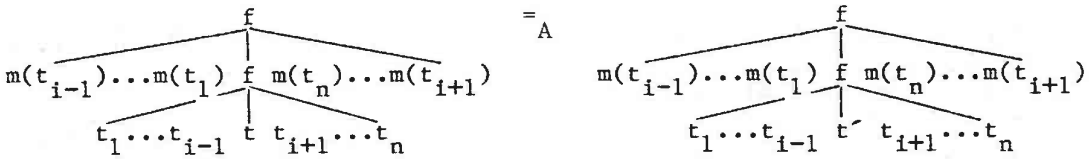
en utilisant l'implication directe.

Les cas où f est le symbole - et où f appartient à F'_1 se traitent de manière identique.

* Sinon, pour un i fixé,

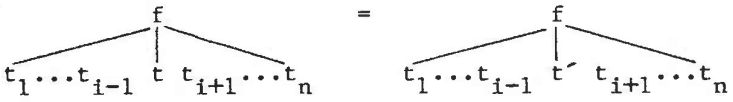


en utilisant l'implication directe:



d'où $t =_A t'$. []

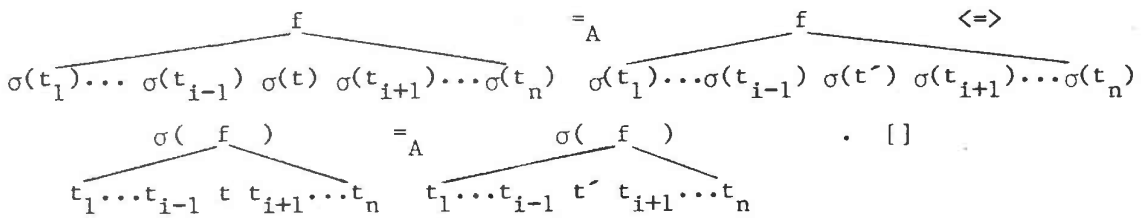
PROPOSITION 3.1: Pour tout symbole f d'arité n non nulle, et pour tous termes $t_1, \ldots, t_n$, t, t', les équations

(t = t') et ()

sont équivalentes pour tout i = 1,...,n.

Preuve: σ étant une substitution,

$\sigma(t) =_A \sigma(t') <=>$



Ces manipulations sur les équations permettent "d'isoler" un sous arbre quelconque dans une équation donnée, comme le résultat suivant le montre de facon constructive:

PROPOSITION 3.2: Pour tous termes t et t', pour tout sous-terme u de t, il existe un terme v tel que l'équation (t = t') soit équivalente à l'équation (u = v).

Preuve: par récurrence structurelle sur t.

* Si t appartient à $F_0 \cup V$, alors t = u et v = t'.

* Si $t = f(t_1)$ avec f appartenant à F_1 par exemple, l'équation (t=t') est équivalente à l'équation $(f\# = f\#)$ donc à l'équation $(t_1 = f\#)$.
  $t \quad t'$ $\qquad$ t'

  Comme u est sous arbre de t_1 par hypothèse, cette dernière équation est équivalente à une équation (u = v).

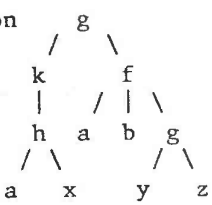
  Il est clair qu'un raisonnement analogue est valable si f est le symbole - ou si f appartient à F'_1.

* Si $t = f(t_1,\ldots,t_n)$ avec par exemple u sous-terme de t_i, alors les équations (t=t') et $(t_i =$ f $)$ sont équivalentes
  $m(t_{i-1})\ldots m(t_1)\ t'\ m(t_n)\ldots m(t_{i+1})$

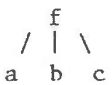
  par la proposition 3.1 et en appliquant l'hypothèse de récurrence, cette dernière est équivalente à une équation (u=v). []

Remarquons que cette preuve est constructive et fournit donc un algorithme de calcul de v.

EXEMPLE : Dans l'équation

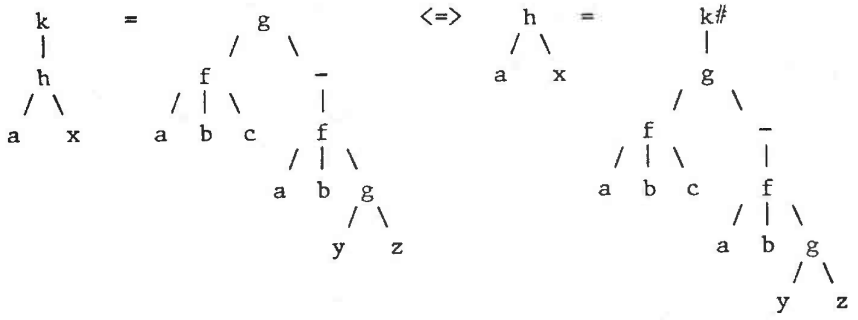


$=$



(1) "isolons" h

(1) $\Leftrightarrow$ $k = t_m(f(a,b,c), g(y,z))$ $\Leftrightarrow$ $h = k\#(f(a,b,c), g(y,z))$



Avec la même preuve, mais en utilisant le lemme 3.2 à la place de la proposition 3.1, on obtient:

COROLLAIRE 3.2: Pour tous termes t et t' et pour tout sous-terme u de t, il existe un terme v tel que $t =_A t'$ si et seulement si $u =_A v$.

REMARQUE: Si m est l'occurrence du sous-terme u dans t, il est facile de vérifier que t' est le sous-terme de v ayant la même occurrence m. Ainsi peut-on écrire v sous la forme $v_m(t')$.

(Rappelons que v_m est une application de AS dans AS, définie par:

pour tout u', $v_m(u') = v[m<- u']$).

Montrons alors que v_m est l'inverse de t_m modulo les axiomes.

* en effet, pour tout terme u' de AS, $t_m(u') =_A t_m(u')$ si et seulement si $u' =_A v_m(t_m(u'))$, à cause du corollaire 3.2.

* d'autre part, pour tout terme u' de AS, $t_m(v_m(u')) =_A u'$ équivaut à $v_m(u') =_A v_m(u')$ toujours à cause du même corollaire.

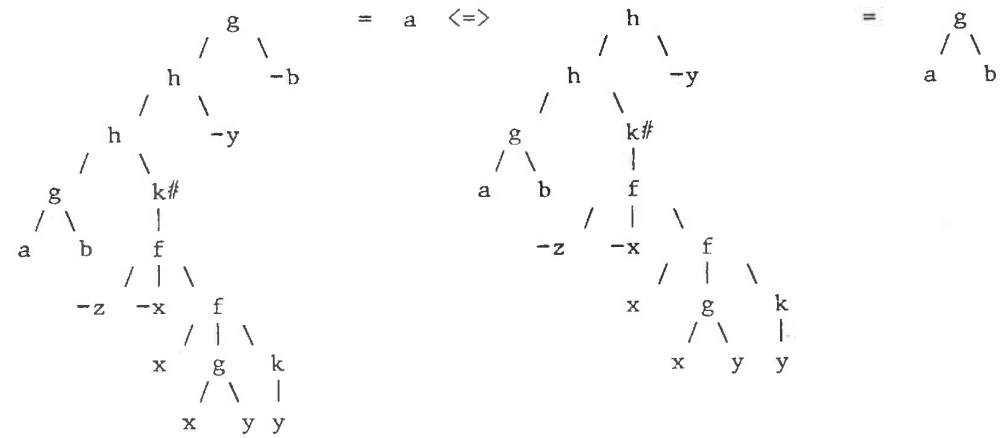
Cette remarque justifie la notation suivante:

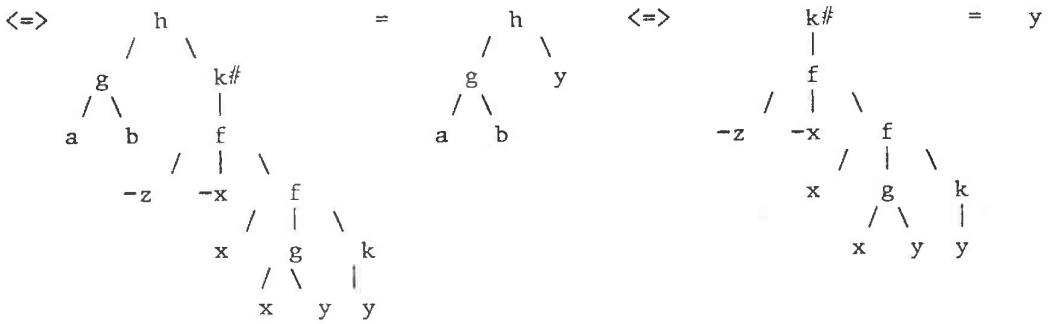
NOTATION: Dans la proposition 3.2, on notera $(u = t_m^{-1}(t'))$ l'équation obtenue en isolant u et équivalente à l'équation $(t=t')$.

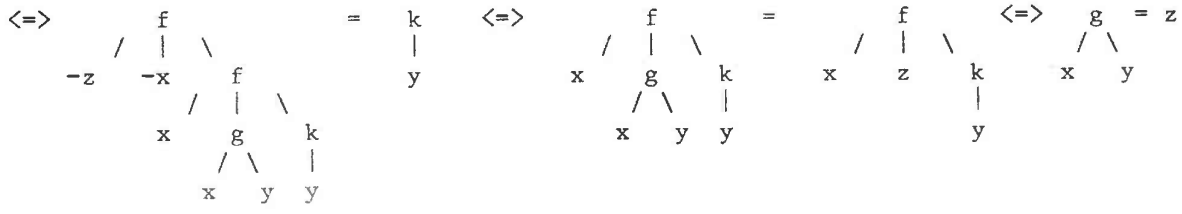
3.2- SIMPLIFICATION D'EQUATIONS

On étudie dans ce paragraphe l'ensemble des équations sur les termes de AS. L'objectif est de donner une méthode de simplification (simplifier une équation étant diminuer sa taille comme nous allons le préciser plus loin), afin de permettre éventuellement une résolution plus facile.

En effet, des équations assez complexes peuvent êtres équivalentes à d'autres beaucoup plus simples, comme dans l'exemple suivant:

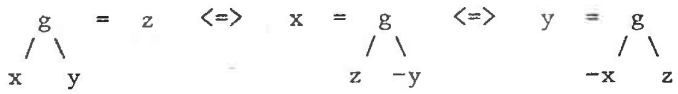






L'équation irréductible (comme on peut s'en convaincre facilement) que l'on

obtient n'est pas unique; en effet:



3.2.1- AXIOMATISATION PARTIELLE DE LA NOTION D'EQUATIONS EQUIVALENTES

Pour axiomatiser partiellement l'équivalence entre équations, nous allons raisonner dans la FU{=}-algebre libre engendrée par V et y définir deux relations de réécriture:

|---> simplifie une équation en une équation équivalente de taille inférieure

|=> réécrit une équation en une équation équivalente dont la taille du membre gauche a diminué, et celle du membre droit augmenté.

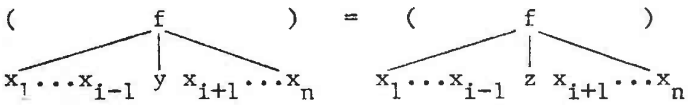
DEFINITION 3.1: Soit Eq l'ensemble des équations $(t_1=t_2)$ avec t_1 et t_2 éléments de AS. Les éléments de Eq sont appelés termes équationnels.

Nous noterons $m(t_1=t_2)$ le terme équationnel $(m(t_2)=m(t_1))$.

Si e est l'équation $(t_1=t_2)$, V(e) désigne $V(t_1)$ U $V(t_2)$.

Pour toute variable x, le nombre d'occurrences de x dans e, noté #(x,e), est égal à $\#(x,t_1) + \#(x,t_2)$.

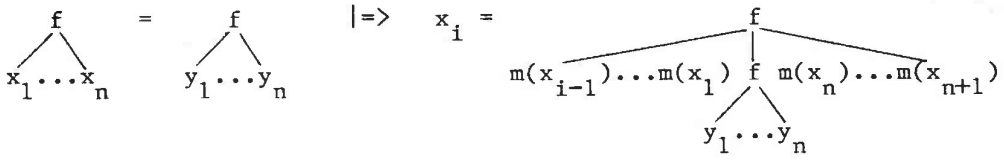
DEFINITION 3.2: A chaque symbole f de F d'arité n non nulle, on associe les N règles de réécriture:



On ajoute la règle (x = x) $|\xrightarrow[0]{}$ ∅, où ∅ est l'éuation trivialement vérifiée.

Le nombre de règles est donc $1*|F_1|+\ldots+k*|F_k|+\ldots+n*|F_n|+1$, si n est l'arité la plus grande de symboles de F et $|F_i|$ le nombre d'éléments de F_i.

On définit alors la relation |=> par les règles de réécriture:

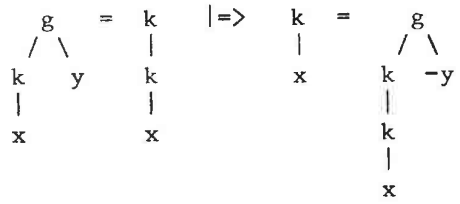


pour chaque symbole f d'arité n non nulle et pour tout i = 1,...,n.

Soit |=| la fermeture réflexive de |=> et <=| l'inverse de |=>.

$|\overset{n}{=}|$ désignera n applications de la relation |=|.

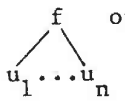
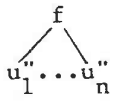
Par exemple

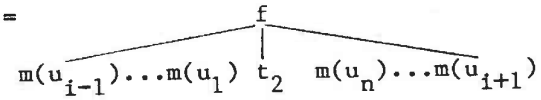


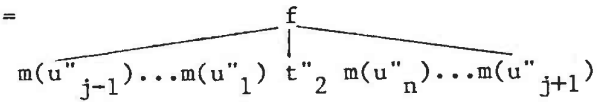
Quelques lemmes techniques sont nécessaires pour dégager deux propriétés de la relation |=> utiles par la suite.

LEMME 3.3: Si $(t_1=t_2)$ |=> $(t'_1=t'_2)$ <=| $(t''_1=t''_2)$ alors

soit $(t''_1=t''_2) = (t_1=t_2)$ soit $(t''_1=t''_2)$ |=| $m(t_1=t_2)$

Preuve: en supposant que t_1 =  on a donc t''_1 = 

Par conséquent on a $t'_1=u_i$ et t'_2 = 

ainsi que $t'_1=u''_j$ et t'_2 = 

* Si i=j, alors $u_k=u''_k$ et $v_k=v''_k$ pour tout k = 1,..,n et $t_2=t''_2$, donc $t''_1=t_1$.

* Si i≠j, par exemple i < j, on doit alors avoir

$m(u_{i-1}) = m(u''_{j-1})$

...

$m(u_1) = m(u''_{j-i+1})$

$m(u''_{j-i}) = t_2$

$m(u''_{j-i-1}) = m(u_n)$

...

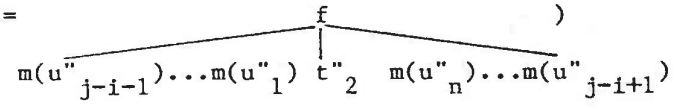
$m(u"_1) = m(u_{n-j+i+2})$

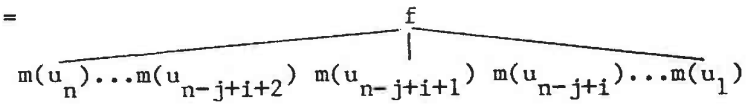
$m(u_{n-j+i+1}) = t"_2$

$m(u"_n) = m(u_{n-j+i})$

...

$m(u"_{j+1}) = m(u_{i+1})$

D'où $(t"_1=t"_2) \mid\Rightarrow (u"_{j-i} =$  $)$

soit: $(m(t_2) =$  $)$

$= (m(t_2)=m(t_1)) = m(t_1=t_2)$. []

LEMME 3.4: Si $(t_1=t_2) \mid\overset{n}{=}\mid (t'_1=t'_2)]$ alors $m(t_1=t_2) \mid\overset{n}{=}\mid m(t'_1=t'_2)$.

Preuve: par récurrence sur n.

* Pour n = 0, le résultat est clair.

* Le cas n = 1 résulte de la définition de m et de $\mid=\mid$.

(Ce cas est utilisé ensuite).

* Prouvons le pour n+1, sachant que cela est vrai jusqu'à n.

$(t_1=t_2) \mid\overset{1}{=}\mid (t"_1=t"_2) \mid\overset{n}{=}\mid (t'_1=t'_2)$

et on applique deux fois l'hypothèse de récurrence. []

3.2.2- CARACTERISATION DES EQUATIONS SIMPLIFIABLES

On note R l'ensemble des termes équationnels réductibles pour $\mid\!\!-\!\!-\!\!-\!\!>$ et dont chaque membre est en forme normale.

DEFINITION 3.3: Un terme équationnel e sera dit simplifiable si et seulement si il existe une suite de termes équationnels tels que:

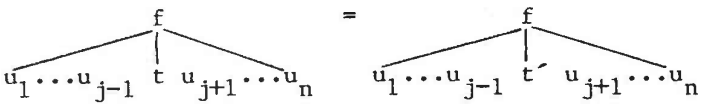
$$e = (t_1^0=t_2^0) \mid\Rightarrow (t_1^1=t_2^1)\ldots\mid\Rightarrow (t_1^p=t_2^p)$$

avec $(t_1^p=t_2^p) \in R$ et pour tout i=0,...,p-1: $|t_1^i| > |t_2^i|$.

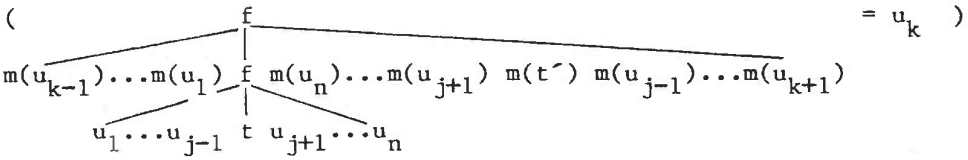
LEMME 3.5: La suite définissant un terme simplifiable est unique.

Preuve: par récurrence sur la longueur de la suite.

* Si $(t_1 = t_2) \overset{1}{|\Rightarrow} e'$, $e' \in R$ avec e' de la forme suivante

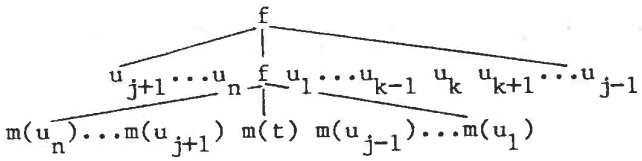


alors $(t_1 = t_2)$ est de la forme



Soit e" une autre équation telle que $(t_1 = t_2) \mid\Rightarrow e"$.

-- si e" a pour membre de gauche m(t') alors son membre de droite n'est pas en forme normale puisque c'est:



-- si par contre le membre de gauche de e" est $m(u_1)$, l'équation $e" = (t'_1 = t'_2)$ vérifie alors $|t'_1| < |t'_2|$ puisque $|t'_1| = |m(u_1)|$ et que t'_2 a pour sous-terme u_1.

On obtient donc encore une contradiction.

En conclusion, il n'y a qu'une possibilité: $e' = e"$.

* Si $(t_1 = t_2)$ est simplifiable en n+1 étapes:

$e = (t_1 = t_2) \overset{1}{|\Rightarrow} (t_1^1 = t_2^1) \overset{n}{|\Rightarrow} (t_1^{n+1} = t_2^{n+1}) = e'$, $e' \in R$.

Soit $e \mid\Rightarrow (t'_1 = t'_2) \mid\Rightarrow \ldots \mid\Rightarrow e_2$, $e_2 \in R$ une autre suite simplifiant e.

Si les deux suites sont distinctes, alors

. si la deuxième suite est de longueur 1:

$(t'_1 = t'_2)$ est réductible; auquel cas, d'après la première partie de la preuve, la suite est unique et n = 0.

. sinon t_1 est de la forme $g(..., u, ...)$ avec $u = t_1^1$.

Toute application de la relation |=> à $(t_1=t_2)$ ne donnant pas $(t_1^1=t_2^1)$ rendra une équation $(t'_1=t'_2)$ telle que $|t'_1| < |t'_2|$.

En effet tout fils de g différent de u est de taille inférieure à u car $|t_1^1| > |t_2^1|$); or, puisque la seconde suite est de longueur supérieure à 1, on a $|t'_1| > |t'_2|$, ce qui fournit une contradiction.

Donc $(t_1^1=t_2^1) = (t'_1=t'_2)$.

En appliquant l'hypothèse de récurrence à $(t_1^1=t_2^1)$, on obtient le résultat. []

Nous sommes alors en mesure de caractériser une équation simplifiable comme étant équivalente à une équation de R. Ce résultat est à la base du processus de simplification d'une équation que nous donnons par la suite.

PROPOSITION 3.3: Soient t_1 et t_2 des termes en forme normale de AS tels que $|t_1| > |t_2|$.

Alors l'équation $(t_1=t_2)$ est équivalente (au sens $|\overset{*}{=}|$) à une équation e appartenant à R si et seulement si il existe une suite unique de termes équationnels :

$$(t_1 = t_2) \;|=>\; (t_1^1 = t_2^1) \ldots |=> \; (t_1^p = t_2^p) \text{ avec } (t_1^p = t_2^p) \in R$$

et pour tout i compris entre 0 et p-1, $|t_1^i| > |t_2^i|$.

Preuve: La réciproque étant évidente, la seule difficulté est de construire une telle suite.

Soit T la suite cherchée. Son unicité découle du lemme 3.5 et il suffit donc de prouver son existence. Considérons pour cela la suite S:

$$(t_1 = t_2) = (s_1^0 = s_2^0) \;|=|\; (s_1^1 = s_2^1) \ldots |=| \; (s_1^n = s_2^n) = e.$$

et raisonnons par récurrence sur la longueur n de S.

* Si n = 0, le résultat est clair.

* Supposons le résultat prouvé avec n et posons:

$(t_1 = t_2) \ |\overset{1}{=}| \ (s_1^1 = s_2^1) \ |\overset{n}{=}| \ e$, $e \in R$

** CAS 1 : $(t_1 = t_2) <=| (s_1^1 = s_2^1) \ |\overset{n}{=}| \ e$, $e \in R$.

L'hypothèse $|t_1| > |t_2|$ implique $|s_1^1| > |s_2^1|$;

donc par hypothèse de récurrence, il existe une suite

$(s_1^1 = s_2^1) \ |=> (t_1^1 = t_2^1) \ldots |=> (t_1^p = t_2^p)$

dont le dernier terme appartient à R.

Supposons que $(t_1 = t_2) =$ ($f(u_1 \ldots u_n) = g(v_1 \ldots v_m)$) et par conséquent

on a $(s_1^1 = s_2^1) =$ ($g(m(v_{i+1}) \ldots m(v_1), f(u_1 \ldots u_n), m(v_m) \ldots m(v_{i-1})) = v_i$)

donc .soit $(t_1 = t_2) =$ ($f(u_1 \ldots u_n) = g(v_1 \ldots v_m)$) et on a trouvé une

suite répondant à la question,

.soit $(t_1^1 = t_2^1) = (m(v_j) =$ $g(v_{j+1} \ldots v_m, f(v_1 \ldots v_i \ldots v_{j-1}), m(u_n) \ldots m(u_1))$)

ce qui est impossible car, en revenant aux définitions de t_2 et de v_j,

t_2 ne serait pas un terme en forme normale.

** CAS 2 : $(t_1 = t_2) \ |=> (s_1^1 = s_2^1) \ |\overset{n}{=}| \ e$, $e \in R$.

Nous avons à nouveau trois cas:

* Soit il existe i tel que S soit de la forme:

$(t_1 = t_2) \ |=> (s_1^1 = s_2^1) \ldots |=> (s_1^{i-1} = s_2^{i-1}) \ |=> (s_1^i = s_2^i) <=| (s_1^{i+1} = s_2^{i+1})$

Or, par le lemme 3.4, soit $(s_1^{i-1} = s_2^{i-1}) = (s_1^{i+1} = s_2^{i+1})$ et $(t_1 = t_2) \ |\overset{n-1}{=}| \ e$

soit $(s_1^{i+1} = s_2^{i+1}) \ |\overset{1}{=}| \ m(s_1^{i-1} = s_2^{i-1})$ et $(t_1 = t_2) \ |\overset{n}{=}| \ m(e)$.

Dans les deux cas, il suffit d'appliquer l'hypothèse de récurrence.

* Soit S est la suite :

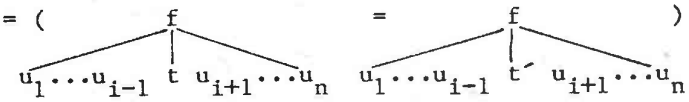
$(t_1 = t_2) \mid\Rightarrow (s_1^1 = s_2^1) \ldots \mid\Rightarrow (s_1^n = s_2^n)$ avec $|s_1^n| > |s_2^n|$

alors pour tout i compris entre 0 et n, $|s_1^i| > |s_2^i|$ et il suffit de prendre dans ce cas T = S.

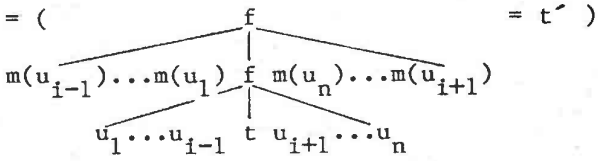
* Soit S est la suite

$(t_1 = t_2) \mid\Rightarrow (s_1^1 = s_2^1) \ldots \mid\Rightarrow (s_1^n = s_2^n)$ avec $|s_1^n| \leq |s_2^n|$.

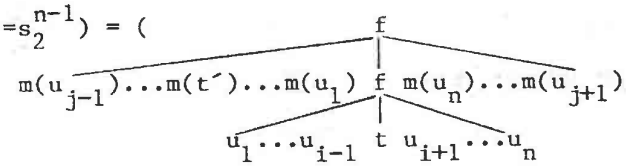
Supposons par exemple que

$(s_1^n = s_2^n) = ($  $)$

Alors,

.soit $(s_1^{n-1} = s_2^{n-1}) = ($  $= t´)$

et c´est impossible car s_1^{n-1} doit être en forme normale.

.soit $(s_1^{n-1} = s_2^{n-1}) = ($  $= u_j)$

et T = S car $|s_1^{n-1}| > |s_2^{n-1}|$ et donc pour tout i de 0 à n, $|s_1^i| > |s_2^i|$.

Ce qui termine la preuve . []

3.2.3- EQUATION IRREDUCTIBLE OBTENUE PAR SIMPLIFICATION

Nous allons maintenant préciser ce que nous entendons par simplifier une équation.

DEFINITION 3.4: Etant donnée une équation e = $(t_1=t_2)$ avec t_1 et t_2 en forme normale pour ECR, on note C(e) l'ensemble des équations équivalentes à e modulo l'équivalence engendrée par |---> et |=>:

$C(e) = \{e' \in Eq \mid e' \ (\ |\overset{*}{\text{---}}\!> . <\overset{*}{\text{---}}\!| . |\overset{*}{=}\!> . <\overset{*}{=}\!|)^* \ e \}$.

Une équation e est minimale ou irréductible si et seulement si il n'existe pas d'équation e' telle que $e \ (|\overset{*}{=}| . |\text{---}\!> . |\overset{*}{=}|)^+ e'$.

Simplifier e , c'est trouver dans C(e) une équation minimale, qui sera évidemment plus facile à résoudre que e.

Pour simplifier une équation e, nous disposons d'un processus algorithmique dont le principe est basé sur le résultat de la proposition 3.3. On échange éventuellement les membres de l'équation e de manière à ce que la taille du membre gauche soit supérieure ou égale à celle du membre droit, puis on réécrit l'équation en lui appliquant la relation |=>, jusqu'à obtenir

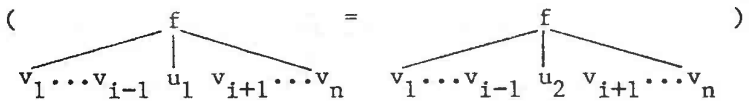
- soit une équation réductible de R, que l'on réduit par la relation |---> en e' sur laquelle on réitère le procédé,
- soit une équation dont le membre gauche est strictement inférieur en taille au membre droit, auquel cas l'équation e est irréductible.

PROPOSITION 3.4: Etant donnée une équation e = $(t_1=t_2)$ avec $|t_1| \geq |t_2|$ t_1 et t_2 en forme normale pour ECR, l'algorithme REDUIRE suivant calcule un élément minimal de C(e):

REDUIRE (t_1=t_2)

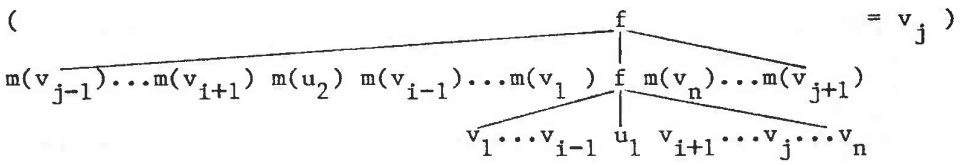
CAS (0) t_1 est un atome et t_1=t_2 ALORS ∅

(1) e est de la forme

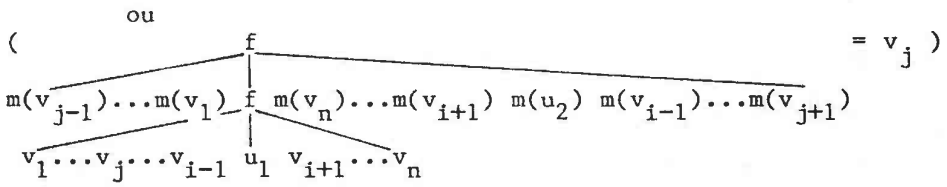


ALORS REDUIRE (u_1=u_2)

(2) e est de la forme



ou

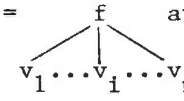


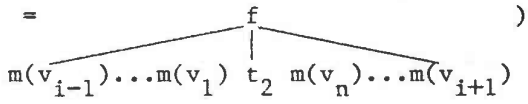
ALORS SI $|u_1| > |u_2|$ ALORS REDUIRE (u_1 = m(u_2))

SINON REDUIRE (m(u_2) = u_1)

FSI

(3) $|t_1| \leq |t_2|$ ALORS (t_1=t_2) est le résultat

(4) t_1 =  avec $|v_i| \geq |v_j|$ pour tout j de 1,...,n différent de i

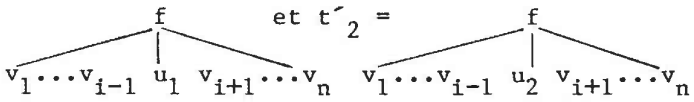
ALORS REDUIRE (v_i = )

FIN CAS

Preuve: REDUIRE termine puisque la taille du terme de gauche est strictement décroissante à chaque appel.

On va prouver que le résultat (t'_1=t'_2) est minimal dans C(e).

Tout d'abord il est clair que (t'_1=t'_2) est dans C(e); supposons qu'il soit réductible. Puisque $|t'_1| \leq |t'_2|$:

* soit $|t'_1|=|t'_2| \geq 1$. La seule simplification possible dans ce cas exige que t'_1 =  et t'_2 =

ce qui est impossible car cela a été vérifié par l'étape (1).

* soit t'_1 et t'_2 sont des atomes et $t'_1=t'_2$, ce qui est impossible car cela a été vérifié à l'étape (0).

* soit $|t'_1| < |t'_2|$. Alors l'étape précédente est une étape (4) dont l'équation $(t"_1=t"_2)$ avec $|t"_1| > |t"_2|$ est réductible si $(t'_1=t'_2)$ est réductible.

Donc $(t"_1=t"_2) \mid\overset{n}{=}\mid e$, $e \in R$ et d'après la proposition 3.3, il existe une suite de termes équationnels unique T:

$(t"_1=t"_2) \mid=> (t_1^1=t_2^1)\ldots\mid=> (t_1^p=t_2^p)$ avec $|t_1^i| > |t_2^i|$ pour tout i de $1,\ldots,p-1$ et $(t_1^p=t_2^p)$ appartenant à R.

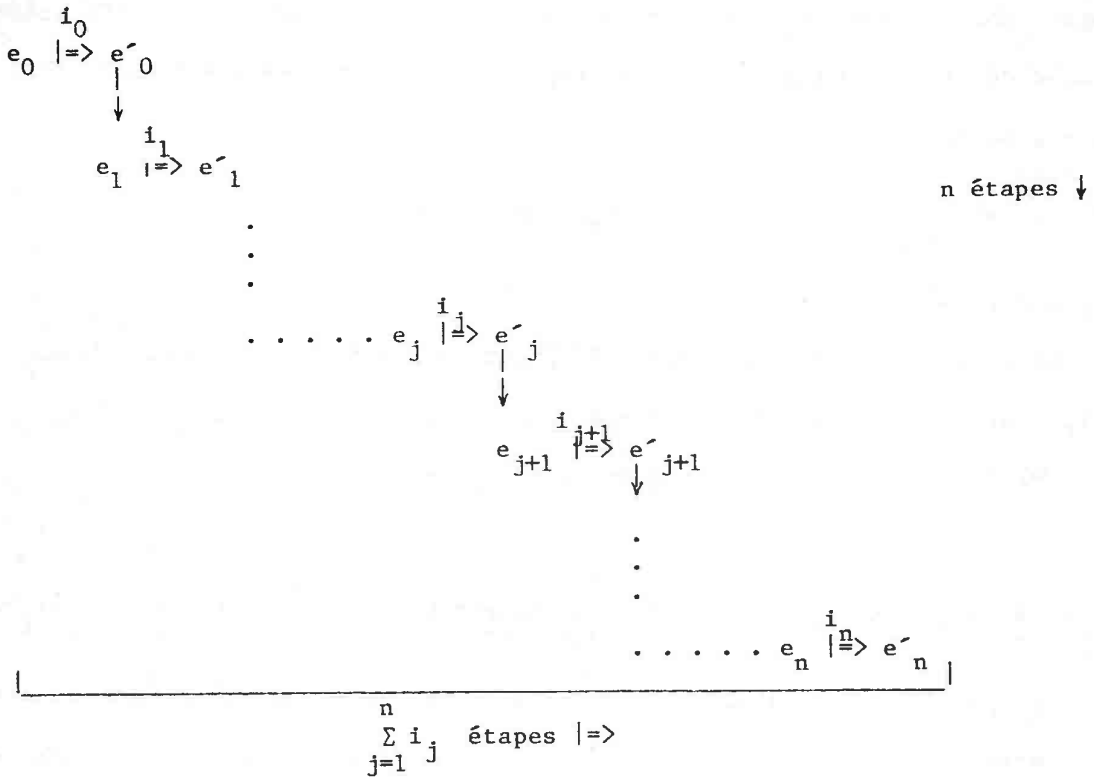
Puisque $(t"_1=t"_2) \mid=> (t'_1=t'_2)$ et à cause de l'unicité de T, on a donc

$(t'_1 = t'_2) = (t_1^1 = t_2^1)$.

Si p>1, alors $|t'_1| > |t'_2|$, ce qui est impossible par hypothèse, donc p=1; mais alors $(t'_1=t'_2)$ appartient à R, ce qui est impossible puisque les étapes (1) et (2) ont été faites avant (3). []

Pour terminer l'étude de cet algorithme, nous allons donner maintenant une évaluation de son coût et montrer qu'il est proportionnel au carré de la taille de l'équation.

Soient e_0, e'_0, e_1, e'_1,..., e_n, e'_n la suite d'équations apparaissant durant l'évaluation de l'algorithme suivant le schéma suivant:



Pour passer de e_j à e'_j, il faut i_j étapes |=> et donc le sous terme simplifié au cours de l'étape de réduction sera de taille supérieure à i_j.

Donc $|e_{j+1}| < |e_j| - 2*i_j$ (a)

D'autre part, supposons que e_j |=> ... |=> e''_j |=> e'_j.

Comme $e''_j = (t'_1 = t''_2)$ vérifie $|t''_1| > |t''_2|$,

$|t''_2| < |e''_j| / 2 = |e_j| / 2$, puisqu'on ne diminue pas la taille de l'équation au cours des étapes |=>.

De plus, le nombre de sous-termes transférés de gauche à droite, c'est-à-dire le nombre d'étapes |=>, est inférieur à $|t''_2| + 1$.

Donc $|e_j| > 2*|t''_2|$, ce qui implique $|e_j| > 2*i_j - 1$. (b)

En combinant les relations (a) et (b) on obtient:

$$2*i_n - 1 < |e_n| < |e_{n-1}| - 2*i_{n-1} < \ldots < |e_0| - 2*\sum_{j=0}^{n-1} i_j$$

D'où $2*\sum_{j=0}^{n} i_j < |e_0| + 1$.

Pour les étapes de réduction |--->, on a clairement $2*n < |e_0|$, car chaque

étape réduit d'au moins deux le nombre de symboles de fonctions de l'équation. Le coût $c(e_0)$ de l'algorithme étant proportionnel au nombre d'étapes |---> et |=>, on aura

$$c(e_0) = k*(n+ \sum_{j=0}^{n} i_j) < k*(|e_0|/2 + |e_0|/2 + 1/2) = k*(|e_0| + 1/2)$$

où k est le coût maximal des étapes |---> et |=>.

Le coût des tests des étapes (1),(2) ou (4) de l'algorithme est borné par $N*|e_0|$, si N est l'arité maximale des symboles de fonctions de e_0.

Par conséquent, $c(e_0)$ est majoré par $|e_0|^2$.

3.2.4- RELATION ENTRE LES EQUATIONS IRREDUCTIBLES OBTENUES PAR SIMPLIFICATION

A partir d'une équation e, nous avons déjà remarqué qu'il existe plusieures équations réduites dans C(e). Il est facile de voir qu'elles sont équivalentes au sens $|\overset{*}{=}|$.

En effet, soient e_1 et e_2 deux équations irréductibles de C(e). Il existe donc une suite finie de termes équationnels

$$e_1 >--> e'_1 >--> \ldots >--> e_p >--> e_2$$

où >--> est soit |---> soit <---| soit |=> soit <=|.

* S'il existe un entier i tel que >--> = |---> alors e_1 est réductible.

* S'il existe un entier j tel que >--> = <---| alors e_2 est réductible.

Dans les deux cas c'est impossible par hypothèse et par conséquent, pour tout i, >--> est une étape |=> ou <=|; donc $e_1 \ |\overset{*}{=}| \ e_2$.

De ce fait, on déduit:

COROLLAIRE 3.3: Pour toutes équations irréductibles e_1 et e_2 de C(e), $V(e_1) = V(e_2)$ et, pour toute variable x, $\#(x,e_1) = \#(x,e_2)$.

REMARQUE: Il est à noter que l'algorithme de simplification d'une équation que nous avons décrit est en fait un algorithme de calcul d'une forme normale

modulo $|\overset{*}{=}|$ pour le système de réécriture |--->, qui a la propriété de $|\overset{*}{=}|$-confluence.

3.3- EQUATIONS LINEAIRES

DEFINITION 3.5: Nous dirons qu'un terme en forme normale t est linéaire en la variable x si et seulement si #(x,t) = 1.

De même une équation e sera linéaire en x si et seulement si #(x,e) = 1.

D'après le paragraphe précédent, une équation non linéaire peut être équivalente à une équation linéaire. La notion de linéarité d'une équation peut donc être étendue à l'ensemble C(e) tout entier.

DEFINITION 3.6: Une équation e étant donnée, on dira que C(e) est linéaire en la variable x si et seulement si un élément irréductible de C(e) est linéaire en x.

Remarquons que le corollaire 3.3 assure la cohérence de cette définition puisque, si une équation irréductible est linéaire, alors toutes les autres équations irréductibles sont aussi linéaires en les mêmes variables.

PROPOSITION 3.5: Soit (t=t') une équation linéaire en une variable x d'occurrence m dans t par exemple, et ($x = t_m^{-1}(t')$) l'équation équivalente. Alors $\{ \sigma = (x \dashrightarrow t_m^{-1}(t')) \}$ est un ensemble minimal complet de A-unificateurs de t et t'.

Preuve : Vérifions les trois conditions de la définition d'un ensemble minimal complet de A-unificateurs.

* σ unifie x et $t_m^{-1}(t')$, donc t et t'.
* toute substitution σ' unifiant t et t' dans AS unifie également x et $t_m^{-1}(t')$. Pour toute variable z de V(t) U V(t')

$\sigma'(\sigma(z)) =_A \sigma'(z)$: en effet

$\sigma'(\sigma(x)) = \sigma'(t_m^{-1}(t')) =_A \sigma'(x)$

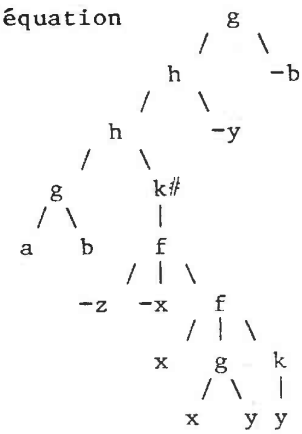
et si z est différent de x, $\sigma'(\sigma(z)) = \sigma'(z)$.

* la troisième condition de la définition d'un ensemble minimal complet de A-unificateurs est trivialement vérifiée. []

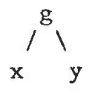
Il est clair qu'il existe autant d'ensemble minimaux complets de A-unificateurs que de variables de l'équation (t=t') n'ayant qu'une seule occurrence dans cette équation. Le résultat précédent permet de prouver que si σ_1 et σ_2 sont les uniques éléments de deux tels ensembles,
$\sigma_1 \leq_A \sigma_2$ et $\sigma_2 \leq_A \sigma_1$ sur V(t)UV(t'), donc $\sigma_1 =_A \sigma_2$ [V(t)UV(t')]

EXEMPLES : (1)- Reprenons un exemple déjà vu.

L'équation



= a se réduit en



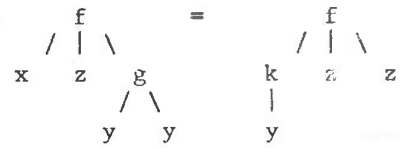
= z

équation linéaire en x, y, z et qui a donc pour ensembles complets minimaux de A-unificateurs:

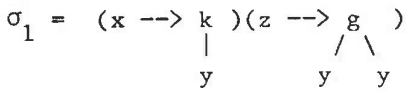
$\{\sigma_1=(z \dashrightarrow$ $)\}$; $\{\sigma_2=(x \dashrightarrow$  $)\}$; $\{\sigma_3=(y \dashrightarrow$ $)\}$.

Il est facile de vérifier que $\sigma_1 =_A \sigma_2 =_A \sigma_3$ sur X={x,y,z}

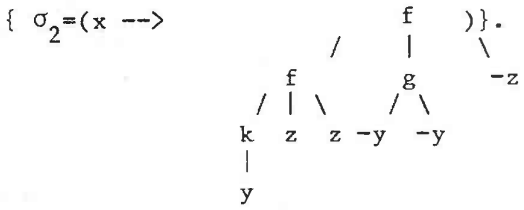
(2)- L'équation



a pour plus petite solution dans l'ensemble des arbres non signés:

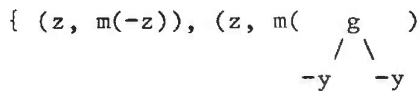


qui n'est pas minimale dans AS, l'ensemble complet minimal de A-unificateurs étant alors, puisque l'équation est linéaire en x:



Il est facile de se convaincre que σ_1 et σ_2 ne sont pas $=_A$-équivalents.

REMARQUE : Partant d'une équation linéaire (t=t') dont les deux membres appartiennent à M(F,V) et sont unifiables, le processus de résolution décrit ci-dessus ne donne pas, en général, l'unificateur minimum σ de t et t' dans M(F,V). Néanmoins, si σ_1 est le A-unificateur trouvé, on a $\sigma_1 \leq_A \sigma$.
Il est légitime de se demander si cette remarque ne permet pas d'obtenir un algorithme d'unification nouveau et performant dans M(F,V).
Une méthode possible est de construire, à partir de $\sigma_1(x)$ un unificande qui serait dans le deuxième exemple ci-dessus:

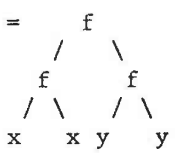


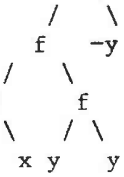
Mais ce procédé ne semble pas conduire à un algorithme performant et nous ne le développerons pas. La question reste donc ouverte.

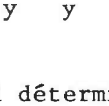
3.4- FILTRAGE DANS LES ARBRES SIGNES

Le problème du filtrage d'un terme t vers un terme t' est également un problème de résolution d'équations, puisqu'il s'agit de trouver une instanciation de t permettant d'obtenir un terme égal à t' dans la théorie.

Quand t est un terme linéaire en une variable x d'occurrence m, les résultats prouvés dans le cas de l'unification permettent de trouver un A-filtre de t vers t': il suffit de renommer toutes les variables de t' par une application bijective ξ , de telle sorte que V(t) et V(ξ(t')) soient disjoints. Il existe alors un A-unificateur $\sigma = (x \dashrightarrow t_m^{-1}(\xi(t')))$ et $\xi^{-1}\sigma$ est un A-filtre de t vers t'. Mais il ne constitue pas en général un ensemble complet, comme on peut le voir sur l'exemple suivant:

Si t =  et t' =  , cette méthode permet de trouver le

A-filtre $\sigma = (x \dashrightarrow$ ) qui n'est pas comparable avec le filtre

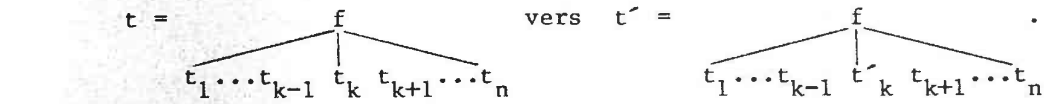
$\alpha = (x \dashrightarrow$)(y --> ).

Nous allons chercher à déterminer dans quels cas il existe un ensemble minimal complet de A-filtres réduit à un seul élément.

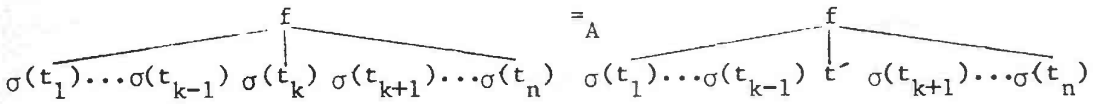
3.4.1- FILTRAGE ET TRANSFORMATIONS D'EQUATIONS

Les transformations d'équations décrites dans la section 3.1 conservent l'ensemble des A-unificateurs. Nous allons voir qu'il n'en est pas de même pour les A-filtres. Les lemmes suivants précisent les cas où ils sont conservés.

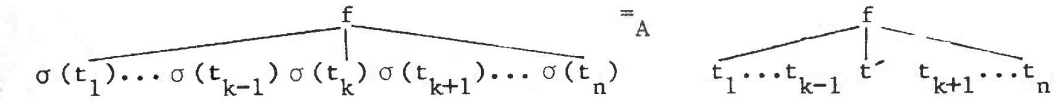
LEMME 3.6: Soit σ une substitution, f un symbole de fonction d'arité n, k un entier compris entre 1 et n, et t'_k, $t_1,\ldots,t_n$ des termes quelconques vérifiant, pour tout i=1,...,n et différent de k, $\sigma(t_j) =_A t_i$. Alors σ est un A-filtre de t_k vers t'_k si et seulement si σ est un A-filtre de



Preuve: $\sigma(t_k) =_A t'_k$ <=>



qui par définition de σ est équivalent à



<=> $\sigma(t) =_A t'$. []

Ce lemme peut s'appliquer dans le cas d'un symbole f d'arité 1; en particulier, pour le symbole -, on obtient:

σ est un A-filtre de t vers t' si et seulement si σ est un A-filtre de -t vers -t'.

LEMME 3.7: Soient t_1,t_2,t'_1,t'_2 des termes tels que les équations $(t_1=t_2)$ et $(t'_1=t'_2)$ soient équivalentes et σ une substitution dont le domaine $D(\sigma)$ est disjoint de $V(t_2)$ et de $V(t'_2)$.
Alors σ est un A-filtre de t_1 vers t_2 si et seulement si σ est un A-filtre de t'_1 vers t'_2.

Preuve: σ étant dans ces conditions un A-unificateur de t_1 et t_2 ainsi que de t'_1 et t'_2, la conclusion est claire. []

Par conséquent, pour deux termes t_1 et t_2:

* Certains A-filtres sont perdus au cours des manipulations.
* Dans le cas d'un terme t_1 linéaire en une variable x, cette variable n'apparait dans aucun des sous-termes de t_1 transférés du membre gauche de l'équation dans le membre droit au cours des transformations isolant x. Donc le premier lemme s'applique à chaque étape et si (x=t) est l'équation équivalente à $(t_1=t_2)$, (x --> t) est un A-filtre de t_1 vers t_2.
* Enfin quand t_1 est linéaire en x n'apparaissant pas dans t_2, l'équation $(t_1=t_2)$ est équivalente à (x=t) et les hypothèses du lemme 3.6 sont vérifiées. Ainsi la substitution (x --> t) est le seul A-filtre de t_1 vers t_2.

Ici encore des hypothèses de linéarité permettent d'obtenir des résultats simples sur l'existence de A-filtres. Nous allons maintenant étudier dans ce cas les ensembles complets et minimaux de A-filtres.

3.4.2- FILTRAGE DANS LE CAS DE TERMES LINEAIRES

Sous certaines conditions, on peut prouver l'existence d'un ensemble minimal complet de A-filtres réduit à un seul élément.

PROPOSITION 3.6: Soient t et t' deux termes, t étant linéaire en x, m l'occurrence de x dans t, et $(x = t_m^{-1}(t'))$ l'équation équivalente à l'équation (t = t').
Si $V(t) \cap V(t')$ est inclus dans {x}, alors $\{\sigma = (x \to t_m^{-1}(t'))\}$ est un ensemble minimal complet de A-filtres de t vers t'.

Preuve: Vérifions les trois conditions de la définition d'un ensemble minimal complet de A-filtres.

* σ est dans F(t,t´), d´après les remarques ci-dessus.

* pour tout A-filtre σ´ de t vers t´, puisque σ´(t) $=_A$ t´,

σ´(t_m^{-1}(t´)) = (σ´(t))$_m$(σ´(x)) $=_A$ t´.

Donc σ´(x) $=_A$ (σ´(t))$_m^{-1}$(t´).

Posons alors σ" = σ´$|_{V(t)\setminus\{x\}}$

et prouvons que pour toute variable z dans V(t), σ"(σ (z)) $=_A$ σ´(z).

** si z est différent de x, σ"(σ (z)) = σ"(z) = σ´(z).

** si z = x, σ"(σ (x)) = σ"(t_m^{-1}(t´)) et

σ´(x) $=_A$ (σ´(t))$_m^{-1}$(t´) = (σ"(t))$_m^{-1}$ (σ"(t´)) = σ"(t_m^{-1}(t´))

D´où $\sigma \leq_A \sigma$´ [V(t)].

* la condition de minimalité est trivialement vérifiée. []

PROPOSITION 3.7: Soient t et t´ deux termes linéaires en x, m l´occurrence de x dans t et (x = t_m^{-1}(t´)) l´équation équivalente à l´équation (t = t´).
Alors { σ = (x --> t_m^{-1}(t´))} est un ensemble minimal complet de A-filtres de t vers t´.

Preuve: elle est semblable à celle de la proposition précédente, excepté en ce qui concerne le deuxième point de la définition, pour lequel il faut faire le raisonnement suivant:

pour tout A-filtre σ´ de t vers t´, cherchons une substitution σ" vérifiant

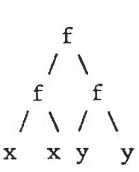
quel que soit z dans V(t), σ"(σ (z)) $=_A$ σ´(z).

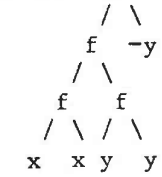
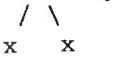
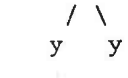
** si z est différent de x, σ"(z) $=_A$ σ´(z)

** si z = x, σ"(t_m^{-1}(t´)) $=_A$ σ´(x).

Il est possible de résoudre ce problème, puisqu´on peut isoler le sous-terme σ"(x) qui apparait à l´occurrence m´ dans t_0= σ"(t_m^{-1}(t´)).
En posant σ"(x) = $(t_0)_{m'}^{-1}$(σ´(x)), la substitution σ" est complètement determinée. []

On peut remarquer que les hypothèses des deux propositions précédentes sont indispensables ainsi que le prouve l'exemple suivant:

$t =$  et $t' =$  ne vérifient pas ces hypothèses et

$\sigma_1 = (x \dashrightarrow$  $) \text{ et } \sigma_2 = (x \dashrightarrow$  $)(y \dashrightarrow$  $)$ ne sont pas comparables.

TRANSITION

Dans le but de résoudre des équations non linéaires sur les arbres signés, nous avons pensé utiliser le processus général de la surréduction qui fournit, dans un bon nombre de théories, un ensemble complet d'unificateurs. Néanmoins dans le cas des arbres signés, l'algorithme qui en est issu ne termine pas, à cause des règles issues des axiomes de (A2). Une idée naturelle, déjà utilisée avec profit dans d'autres cadres, est alors de travailler modulo ces axiomes. Cela nous amènera à introduire une nouvelle notion de surréduction. Dans cette deuxième partie de notre travail, nous nous sommes tout d'abord attachés à décrire cet outil général, la R,E-surréduction, en parallèle avec les propriétés connues de la surréduction. Pour pouvoir l'utiliser, il faut disposer:

d'un algorithme de E-unification complet

d'un système de réécriture R qui soit E-canonique

d'une propriété supplémentaire de E-commutation.

Notre but est donc maintenant de trouver un ensemble d'axiomes E et un système de réécriture R qui soit E-canonique, à partir de l'ensemble A réunion des ensembles d'axiomes suivants, dans la F-algèbre des arbres signés AS:

(A1)= { $--x = x$ }

(A2)= {$-f(x_1,\ldots,x_n) = f(-x_n,\ldots,x_1)$ pour chaque f de F U F'_1}

(A3)= { $f(f\#(x)) = x$ pour chaque symbole unaire f }

(A3')= { $f\#(f(x)) = x$ pour chaque symbole unaire f }

(A4)= { $f(-x_{i-1},\ldots,-x_1, f(x_1,\ldots,x_{i-1}, x, x_{i+1},\ldots,x_n), -x_n,\ldots,-x_{i+1}) = x$ | f symbole d'arité au moins deux de F}

Les choix de E et R sont motivés par les constatations suivantes:

- l'apparition d'une infinité de règles dans l'algorithme de complétion de Knuth et Bendix est due à l'existence des axiomes de (A2), comme nous l'avons remarqué dans le chapitre deux.

- le processus de surréduction peut ne pas terminer dans certains cas à cause de ces mêmes axiomes, comme nous le verrons par la suite dans le chapitre sept.

- d'autre part, le système de réécriture associé aux axiomes (A1)U(A2) est confluent, ce qui permet de décider de la congruence engendrée par ce sous-ensemble des axiomes.

Le choix naturel est alors de partitionner l'ensemble A en E = (A1)U(A2) et R = (A3)U(A3')U(A4). Les axiomes de E engendrent une théorie équationnelle décidable notée $=_E$; les axiomes de R sont orientés de gauche à droite pour constituer un système de réécriture que nous désignerons encore par R. L'égalité dans la théorie, notée $=_A$ n'est autre que $(=_E \cup (\xrightarrow{R}) \cup (\xrightarrow{R})^{-1})^*$

Cela motive

** l'extension de l'algorithme de Martelli et Montanari que nous présentons dans le chapitre cinq, pour pouvoir construire un ensemble complet de E-unificateurs.

** une deuxième étude de la théorie équationnelle engendrée par l'ensemble d'axiomes A dans AS, qui est exposée dans le chapitre six.

L'application de ces résultats à l'algèbre des arbres signés sera présentée dans le chapitre sept.

CHAPITRE QUATRE

UNE METHODE INCREMENTALE D'UNIFICATION DANS LES THEORIES EQUATIONNELLES

INTRODUCTION

Nous nous intéressons dans ce chapitre à un processus général de résolution d'équations dans une théorie équationnelle A, décrit initialement par M.Fay [FAY,79] et amélioré par D.S.Lankford et A.M.Ballantyne [L&B,79], puis J.M.Hullot [HUL,80].

S'il existe un système de réécriture canonique associé à A, ce processus permet de construire un ensemble complet de A-unificateurs de deux termes. Un résultat analogue a été trouvé par G.Huet dans le cas du lambda-calcul typé [HUE,75]. L'algorithme décrit par M.Fay ne donne pas, en général, une procédure de décision de la A-égalité, car sa terminaison n'est pas assurée. Néanmoins on peut en déduire une procédure de semi-décision en ce sens que, si deux termes sont A-unifiables, alors une solution sera trouvée en un temps fini. J.M. Hullot a repris ces résultats en dégageant précisément les propriétés d'une relation appelée surréduction, qui est à la base de la construction de M.Fay. Il a supprimé de l'algorithme décrit par M.Fay certaines redondances et a donné une condition suffisante de sa terminaison [HUL,80].

Nous généralisons ici les résultats obtenus par J.M.Hullot à l'extension de la notion de système de réécriture, dans le cas où la théorie équationnelle A se décompose en un ensemble d'axiomes E et un système de réécriture R. Nous reprenons la définition, proposée par D.S.Lankford et A.M.Ballantyne, puis par J.M.Hullot, d'une relation de R,E-surréduction et nous établissons le lien avec

la R,E-réduction. Ensuite, nous donnons un théorème général permettant de construire un ensemble complet de A-unificateurs par un algorithme non déterministe dont la terminaison n'est pas assurée. Nous montrons ensuite comment éliminer certaines redondances et obtenir également une condition suffisante de terminaison.

4.1- LA SURREDUCTION: UN OUTIL POUR LA RESOLUTION D'EQUATIONS DANS LES THEORIES EQUATIONNELLES

Ce paragraphe contient les définitions et les résultats utilisés dans la suite, concernant la surréduction. Nous les avons empruntés à la thèse de J.M.Hullot et nous y renvoyons le lecteur intéressé par les preuves que nous ne donnos pas.

Dans toute cette section, nous abrégeons $--->^R$ en --->.

4.1.1- DEFINITION DE LA SURREDUCTION

Informellement, surréduire un terme c'est lui appliquer une substitution faisant apparaitre l'application d'une réduction. Nous allons définir tout d'abord la relation de surréduction que nous noterons -V--->.

Nous noterons O(t) l'ensemble des occurrences m appartenant à dom(t) telles que $t_{|m}$ ne soit pas une variable. Nous dirons parfois que m est une occurrence de non variable de t.

DEFINITION 4.1: Soit t un terme et W un ensemble fini de variables contenant V(t). Si $g_k ---> d_k$ est une règle de R, il est toujours possible de renommer les variables de g_k de telle sorte que l'intersection de $V(g_k)$ et de W soit vide et nous supposerons toujours cette condition réalisée. Désignons par W_k la réunion de $V(g_k)$ et de W.

Nous dirons que t se surréduit en t' à l'occurrence m de O(t), avec la règle k et la substitution σ et nous noterons

$$t \text{ -V---> }_{[m,k,\sigma]} t'$$

si et seulement si

* $t_{|m}$ et g_k sont unifiables et σ est l'unificateur minimum de ces deux termes en dehors de W_k .

* $t' = \sigma\ (t[m<-d_k])$

La substitution σ sera dite surréductrice.

Nous appellerons surdérivation toute suite de surréductions issue d'un terme.

Il est clair que tout terme réductible par une règle de R est surréductible avec cette même règle.

EXEMPLE : Considérons le système canonique de réécriture de termes:

$$R = \{\ f(x,a) \text{ ---> } x\ \}$$

où f et a sont des symboles de fonctions.

Le terme f(f(y,z),z) est surréductible à l'occurrence ε en f(y,a) avec la substitution σ =(x-->f(y,a))(z-->a)

et à l'occurrence 1 en f(y,a) également, avec la sustitution σ =(x-->y)(z-->a).

L'idée fondamentale de ce qui suit est la suivante: résoudre l'équation (t=t') dans la théorie A, c'est trouver une substitution σ normalisée vérifiant $\sigma(t) =_A \sigma(t')$ ou de facon équivalente $FN(\sigma(t)) = FN(\sigma(t'))$. Supposons qu'une telle substitution existe: soit elle unifie t et t', auquel cas on peut la trouver par un algorithme d'unification classique, soit elle ne les unifie pas et l'un des deux termes $\sigma(t)$ ou $\sigma(t')$ est alors réductible, faute de quoi il serait impossible d'avoir l'égalité de leur formes normales. Cela revient à dire alors que l'un des deux termes t ou t' est surréductible.

4.1.2- CORRESPONDANCE ENTRE SURREDUCTION ET REDUCTION

Le théorème fondamental suivant précise la remarque précédente en montrant que sur une dérivation issue de $\eta(t)$ (où η est une substitution normalisée) on peut "calquer" une surdérivation issue de t et réciproquement. Nous donnons entièrement la démonstration, volontairement détaillée, de ce résultat dû à J.M.Hullot [HUL,80].

THEOREME 4.1: Soient u un terme, W un ensemble fini de variables contenant V(u), η une substitution normalisée telle que $D(\eta)$ soit inclus dans V(u).
A toute dérivation (1) issue de $\eta(u)$, on peut associer une surdérivation issue de u:

$$\begin{array}{ccccccc} \eta(u)=v_0 & \xrightarrow{\quad} & [m_0,k_0] & v_1 \quad \cdots & \xrightarrow{\quad} & [m_{n-1},k_{n-1}] & v_n \quad (1) \\ \uparrow \eta_0 & & & \uparrow \eta_1 & & & \uparrow \eta_n \\ u=u_0 & \xrightarrow{-V\rightarrow} & [m_0,k_0,\sigma_0] & u_1 \cdots & \xrightarrow{-V\rightarrow} & [m_{n-1},k_{n-1},\sigma_{n-1}] & u_n \quad (2) \end{array}$$

telle que, pour tout i compris entre 0 et n, il existe une substitution η_i et un ensemble de variables V_i vérifiant:

a) $D(\eta_i)$ est inclus dans V_i

b) η_i est normalisée

c) $\eta|_W=(\eta_i\theta_i)|_W$ et $I(\theta_i)$ est inclus dans V_i

où θ_i est défini par récurrence par

$\theta_0=Id$ et $\theta_{i+1}=\sigma_i\theta_i$ pour i compris entre 0 et n-1.

d) $V(u_i)$ est inclus dans V_i et $\eta_i(u_i)=v_i$

Réciproquement, à toute surdérivation (2) issue de u et à toute substitution η vérifiant, avec les notations précédentes, $\theta_n \leq \eta\ [W]$ est associée une dérivation (1) issue de $\eta(u)$.

Preuve: Démontrons la première partie du théorème par récurrence sur l'entier i.

* pour i=0, le résultat est clair en prenant $\eta_0 = \eta$

$$V_0 = W \cup D(\eta) = W$$

$$I(\theta_0) = \emptyset$$

* supposons maintenant a,b,c,d vrais pour i.

Comme $v_i \xrightarrow{[m_i,k_i]} v_{i+1}$, il existe une substitution σ telle que $(v_i)_{|m_i} = \sigma(g_{k_i})$ et $v_{i+1} = v_i[m_i \leftarrow \sigma(d_{k_i})]$.
Rappelons que nous imposons à l'ensemble $V(g_{k_i})$ des variables de g_{k_i} d'être disjoint de V_i, ce qu'il est toujours possible d'obtenir par un renommage des variables.

Puisque, par hypothèse, $\eta_i(u_i) = v_i$ et que η_i est normalisée, m_i appartient à $O(u_i)$ et par conséquent

$$\eta_i((u_i)_{|m_i}) = (\eta_i(u_i))_{|m_i} = (v_i)_{|m_i} = \sigma(g_{k_i}).$$

η_i et σ étant à supports disjoints, on peut considérer la substitution $\rho = \eta_i \cup \sigma$ qui unifie $(u_i)_{|m_i}$ et g_{k_i} d'après ce qui précède.

Il existe donc un unificateur minimum σ_i vérifiant

$$\sigma_i \leq \rho \quad [V_i \cup V(g_{k_i})]$$

et $u_i \xrightarrow{V}{}_{[m_i,k_i,\sigma_i]} u_{i+1} = \sigma_i(u_i[m_i \leftarrow d_{k_i}])$.
D'autre part, il existe une substitution η' telle que

$$\eta' \sigma_i = \rho \quad [V_i \cup V(g_{k_i})]$$

Définissons alors V_{i+1} et η_{i+1} de la facon suivante:

$$V_{i+1} = (V_i \cup I(\sigma_i)) \setminus D(\sigma_i),$$

$$\eta_{i+1} = \eta'_{|V_{i+1}}$$

et vérifions a,b,c,d pour i+1:

a) $D(\eta_{i+1})$ est inclus dans V_{i+1} par définition de η_{i+1}

b) η_{i+1} est normalisée. En effet, pour toute variable x de V_{i+1}:

* soit x appartient à $I(\sigma_i)$ et il existe alors une variable y y appartenant à V_i et à $D(\sigma_i)$ telle que $\sigma_i(y)=x$.

$\eta_{i+1}(x) = \eta_{i+1}\sigma_i(y) = \eta_i(y)$ qui est irréductible puisque η_i estnormalisée par hypothèse.

* soit x n´appartient pas à $I(\sigma_i)$ et $\sigma_i(x)=x$.

Donc $\eta_{i+1}(x) = \eta_{i+1}\sigma_i(x) = \eta_i(x)$ qui est irréductible.

c) montrons que $\eta_{|W} = (\eta_{i+1}\theta_{i+1})_{|W}$.

Par hypothèse de récurrence, $\eta_{|W} = (\eta_i\theta_i)_{|W}$.

Or $\eta_i = (\eta_{i+1}\sigma_i)_{|V_i}$.

Puisque $I(\theta_i)$ est inclus dans V_i

$I(\theta_{i+1}) = I(\sigma_i\ \theta_i) \quad \subseteq I(\sigma_i) \cup I(\theta_i)$ est inclus dans $I(\sigma_i) \cup V_i$ lui-même inclus dans V_{i+1}.

On obtient donc:

$$\eta_{|W} = (((\eta_{i+1}\sigma_i)_{|V_i})\theta_i)_{|W} = (\eta_{i+1}\sigma_i\theta_i)_{|W}$$
$$= (\eta_{i+1}\theta_{i+1})_{|W}.$$

d) Dans le but de montrer que $\eta_{i+1}(u_{i+1}) = v_{i+1}$, prouvons d´abord que $V(u_{i+1})$ est inclus dans V_{i+1} :

$V(u_{i+1})$ est inclus dans $V(u_i) \cup I(\sigma_i)$ par définition de u_{i+1} et du fait que $V(\sigma_i(d_{k_i}))$ est inclus dans $I(\sigma_i)$.

Donc $V(u_{i+1})$ est inclus dans $V_i \cup I(\sigma_i) = V_{i+1}$

On a alors:

$$\eta_{i+1}(u_{i+1}) = \eta_{i+1}\sigma_i(u_i[m_i<-d_{k_i}])$$
$$= (\eta_{i+1}\ \sigma_i(u_i))[m_i<-\eta_{i+1}\sigma_i(d_{k_i})]$$

Or $V(\sigma_i(d_{k_i}))$ est inclus dans V_{i+1} car $\sigma_i(d_{k_i})$ est un sous-arbre de u_{i+1} et donc

$\eta_{i+1}\ \sigma_i(d_{k_i}) = \eta\sigma_i(d_{k_i}) = \sigma(d_{k_i})$.

De plus $\rho = \eta_i \cup \sigma$ est égal à σ sur d_{k_i} puisque $V(d_{k_i})$ est inclus dans $V(g_{k_i})$ et est donc disjoint de V_i,

et $\eta_{i+1}\sigma_i$ est égal à η_i sur V_i. On en déduit:

$$\eta_{i+1}(u_{i+1}) = (\eta_i(u_i))[m_i <- \sigma(d_{k_i})]$$
$$= v_i[m_i <- \sigma(d_{k_i})] = v_{i+1}.$$

La première partie du théorème est donc prouvée.

Réciproquement, supposons données une surdérivation issue du terme u et une substitution η vérifiant $\theta_n \leq \eta$ [W] où θ_n désigne la composition de toutes les substitutions σ_i surréductrices faites au cours de la surdérivation.

Il existe donc une substitution ρ telle que $\rho\,\theta_n = \eta$ [W].

Définissons $\eta_n = \rho$,

$\eta_i = \eta_{i+1}\sigma_i$ pour i compris entre 0 et n-1,

et $v_i = \eta_i(u_i)$ pour i compris entre 0 et n-1.

Montrons par récurrence sur i que

$$\eta(u)=v_0 \ --->_{[m_0,k_0]} \ v_1 \ \ldots \ --->_{[m_{n-1},k_{n-1}]} \ v_n$$

* pour i=0, le résultat est clair.

* supposons le pour i;

$$(v_i)|_{m_i} = (\eta_i(u_i))|_{m_i} = (\eta_{i+1}\sigma_i(u_i))|_{m_i}$$
$$= \eta_{i+1}\sigma_i(d_{k_i}) = \eta_i(d_{k_i}) \quad \text{et donc}$$

v_i est réductible à l'occurrence m_i avec la règle k_i.

D'autre part:

$$v_{i+1} = \eta_{i+1}(u_{i+1}) = \eta_{i+1}\sigma_i(u_i[m_i <- d_{k_i}])$$
$$= \eta_i(u_i[m_i <- d_{k_i}])$$
$$= (\eta_i(u_i))[m_i <- \eta_i(d_{k_i})]$$
$$= v_i[m_i <- \eta_i(d_{k_i})]$$

et donc $v_i \ --->_{[m_i,k_i]} \ v_{i+1}$.

Enfin, $v_0 = \eta_0(u_0) = \eta_0(u) = \eta_n\theta_n(u)$

$= \rho\,\theta_n(u) = \eta(u)$ puisque V(u) est inclus dans W.

Ce qui termine la preuve de la deuxième partie du théorème. []

4.1.3- RELATION ENTRE SURREDUCTION ET UNIFICATION

Nous allons maintenant montrer comment combiner l'unification ordinaire et le processus de surréduction pour construire des A-unificateurs de deux termes t et t'. Nous aurons à itérer le processus sur les deux termes en parallèle et pour cela, nous introduisons un symbole de fonction * n'appartenant pas à l'ensemble des symboles de fonction de l'algèbre, et nous commencons l'itération sur le terme *(t,t').

LEMME 4.1: Considérons une surdérivation issue de *(t,t')

$$u=*(t,t')=u_0 \text{ -V---> } u_1=*(t_1,t'_1) \text{ ... -V---> } u_n=*(t_n,t'_n)$$

telle que t_n et t'_n soient unifiables par une substitution μ .

Alors $\mu\,\theta_n$ est un A-unificateur de t et t'. (Rappelons que θ_n est la composition des substitutions surréductrices faites au cours de la surdérivation).

Montrons maintenant que tout A-unificateur peut être atteint de cette manière. Cela résulte du lemme suivant:

LEMME 4.2: Soient t et t' deux termes A-unifiables, ρ un A-unificateur quelconque et W un ensemble de variables contenant V(t)UV(t').
Alors il existe une surdérivation issue de u=*(t,t')

$$u=*(t,t')=u_0 \text{ -V---> } u_1=*(t_1,t'_1) \text{ ... -V---> } u_n=*(t_n,t'_n)$$

telle que t_n et t'_n soient unifiables. Si μ est leur unificateur minimum,

$$\mu\,\theta_n \leq_A \rho\ [W].$$

De plus on peut choisir cette surdérivation de telle sorte que, pour tout i compris entre 0 et n, $(\theta_i)_{|W}$ soit normalisée.

Nous déduisons de ces deux lemmes la construction d'un ensemble complet de A-unificateurs de deux termes t et t'.

THEOREME 4.2: Soit Σ l'ensemble des substitutions σ vérifiant:

il existe une surdérivation issue de $u=*(t,t')$

$u=*(t,t')=u_0 \text{ -V---> } u_1=*(t_1,t'_1) \text{ ... -V---> } u_n=*(t_n,t'_n)$

telle que t_n et t'_n soient unifiables, $(\theta_n)_{|W}$ soit normalisée,

et $\sigma = \mu\theta_n$ où μ est l'unificateur minimum de t et t'.

Alors Σ est un ensemble complet de A-unificateurs en dehors de W.

Un algorithme non déterministe de A-unification de t et t' consiste à énumérer les éléments de Σ en construisant l'arbre de toutes les surdérivations issues de $*(t,t')$. M.Fay décrit un algorithme analogue [FAY,79] à ceci près qu'il normalise les termes à chaque étape, ce qui n'est pas le cas ici. L'ensemble complet de A-unificateurs énuméré n'est pas minimal et en général contient plusieures fois le même A-unificateur. J.M.Hullot montre comment éliminer certaines redondances en se restreignant à des surdérivations particulières. Il en déduit de plus une condition suffisante de terminaison de l'algorithme pour un type particulier de systèmes canoniques de réécriture [HUL,80].

4.1.4- SURREDUCTION BASIQUE

Définissons d'abord les dérivations et les surdérivations basées sur un ensemble d'occurrences.

DEFINITION 4.2: Soient u et v deux termes tels que $v=\eta(u)$ et $U=O(u)$.

Une dérivation issue de v

$v=v_0 \text{ ---}>_{[m_0,k_0]} v_1 \text{ ... ---}>_{[m_{n-1},k_{n-1}]} v_n$

ou une surdérivation issue de u

$u=u_0 \text{ -V---}>_{[m_0,k_0,\sigma_0]} u_1 \text{ ... -V---}>_{[m_{n-1},k_{n-1},\sigma_{n-1}]} u_n$

est basée sur U si et seulement si

* il existe une suite d'ensembles d'occurrences U_i inclus dans $O(v_i)$

telle que $U_0 = U$

$$U_{i+1} = (U_i \setminus \{p \in U_i \mid m_i \leq p\}) \cup \{m_i.p \mid p \in O(d_{k_i})\}$$

* à chaque étape $m_i \in U_i$

Intuitivement, la suite des U_i, c'est-à-dire l'ensemble des occurrences où l'on s'autorise une réduction ou une surréduction à l'étape i, se construit en ne prenant que les occurrences de sous-termes qui ne sont pas apportés par une substitution réductrice ou surréductrice précédente.

L'existence de dérivations basées est assurée par le lemme suivant, qui nécessite tout d'abord une définition.

DEFINITION 4.3: Une dérivation suit une stratégie de l'intérieur vers l'extérieur si, à chaque étape de réduction $u_i \xrightarrow{[m_i,k_i]} u_{i+1}$, la substitution σ telle que $u_{i+1} = u_i[m_i \leftarrow \sigma(d_{k_i})]$ est normalisée.

LEMME 4.3: Soit $v=\eta(u)$ avec η normalisée. Toute dérivation issue de v et suivant une stratégie de l'intérieur vers l'extérieur est basée sur O(u).

On en déduit:

LEMME 4.4: Soit $v=\eta(u)$ avec η normalisée. Il existe une dérivation issue de v et allant à FN(v) qui est basée sur O(u).

Considérons maintenant les surdérivations associées:

DEFINITION 4.4: Une surdérivation issue de u est basique si et seulement si elle est basée sur O(u).

THEOREME 4.3: Les conclusions de théorème 4.1 restent vraies si on considère uniquement les surdérivations basiques.

L'intérêt de ce résultat est de donner une condition suffisante de terminaison du processus de surréduction basique:

PROPOSITION 4.1: Soit R un système de réécriture canonique tel que toute surdérivation basique issue d'un membre gauche de règle termine. Alors toute surdérivation basique issue d'un terme quelconque termine.

Ce résultat s'applique en particulier si les équations définissant la théorie équationnelle n'ont pas de variables, auquel cas la surréduction fournit une méthode générale de résolution d'une équation quelconque. Mais il s'applique aussi quand toutes les parties droites des règles du système de réécriture canonique sont des variables. C'est le cas si le système est réduit à la règle f(x,x)--->x, ainsi que pour la théorie des quasi-groupes, exemple développé dans [HUL,80].

Malheureusement, de nombreux systèmes de réécriture ne vérifient pas cette condition. C'est par exemple le cas des arbres signés. Les résultats que nous présentons dans la section suivante permettent de travailler avec des systèmes de réécriture ne vérifiant pas les hypothèses de la proposition 4.1, soit parce qu'ils n'ont pas la propriété de terminaison finie ou de confluence, soit parce qu'une surdérivation issue d'un membre gauche de règle ne termine pas. Dans les deux cas, le principe adopté est de considérer une partition des axiomes en règles et en équations. Le choix des axiomes à conserver sous forme d'équations semble être naturellement guidé par la détermination des axiomes qui empêchent un processus de terminer, que ce soit, suivant les cas, une dérivation, une surdérivation ou l'algorithme de complétion de Knuth et Bendix.

4.2- LA R,E-SURREDUCTION: UN PROCESSUS D'UNIFICATION INCREMENTAL

Nous allons maintenant étudier la notion de R,E-surréduction. Elle étend la notion de surréduction aux systèmes de réécriture R qui sont E-canoniques. L'étude de tels systèmes a été présentée dans le premier chapitre. Moyennant des hypothèses appropriées, nous donnons une généralisation des résultats obtenus au paragraphe précédent.

Dans tout ce paragraphe, nous supposerons que E est un ensemble d'axiomes $(g_k=d_k)_{k=1,\ldots,n}$ tels que $V(g_k)=V(d_k)$ et pour lequel on connait un algorithme complet de E-unification.

4.2.1- DEFINITION DE LA R,E-SURREDUCTION

Cette définition est tout-à-fait analogue à celle donnée au paragraphe 4.1 mais en utilisant la E-unification au lieu de l'unification.

DEFINITION 4.5: Soit t un terme et W un ensemble fini de variables contenant les variables de t. S'il existe

* une occurrence m de non variable dans t
* une règle $g_k \text{---}>d_k$ dont les variables ont été renommées de telle sorte que $W \cap V(g_k)=\emptyset$
* un E-unificateur de g_k et $t_{|m}$ en dehors de W_k, W_k étant un ensemble fini de variables contenant W et $V(g_k)$,

alors chaque élément de l'ensemble complet de E-unificateurs Σ de $t_{|m}$ et g_k en dehors de W_k est appellé substitution R,E-surréductrice.

Si t' est le terme $\sigma\ (t[m<-d_k])$ où σ est une substitution R,E-surréductrice, on dira que t est R,E-surréductible en t' à l'occurrence m en utilisant la règle k. Ce que nous noterons

$$t \text{-V---}>^{R,E}_{[m,k,\ \sigma]} t'$$

La relation $\text{-V---}>^{R,E}$ est appellée relation de R,E-surréduction sur

l'ensemble des termes.

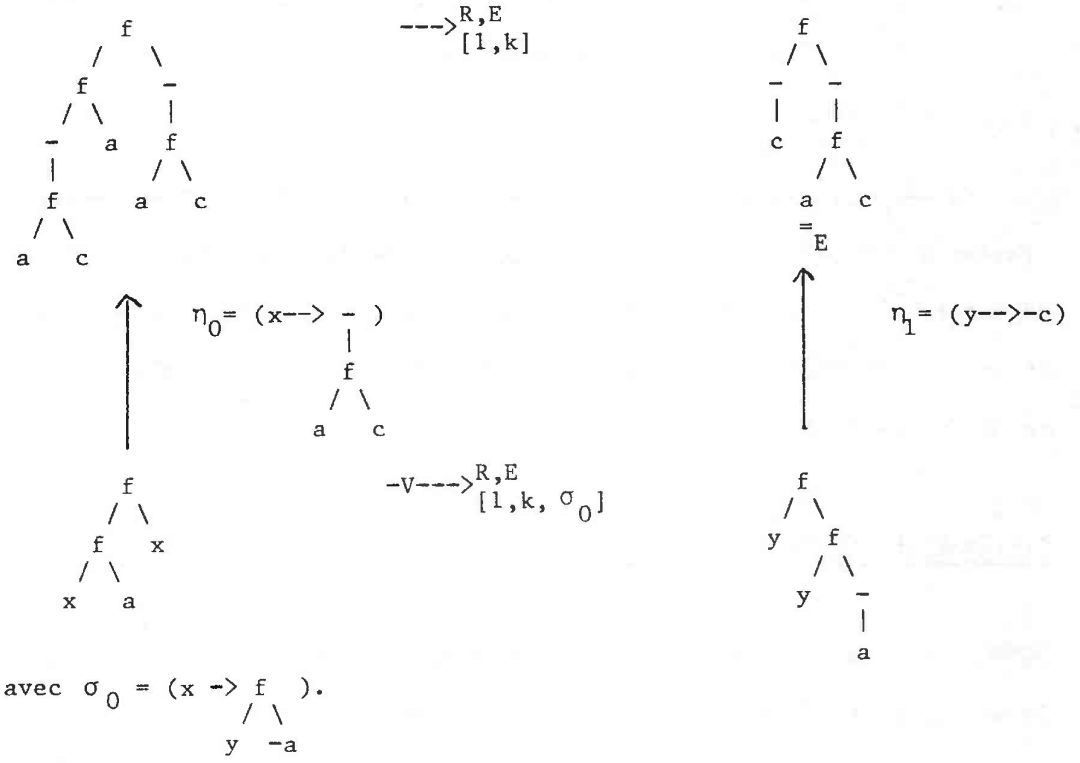
La correspondance établie par le théorème 4.1 entre surréduction et réduction ne se généralise pas sans hypothèse supplémentaire au cas de la R,E-surréduction et de la R,E-réduction. Supposons par exemple que R contienne la règle k

$$f(f(y',-x'),x') \longrightarrow y'$$

et E l'axiome

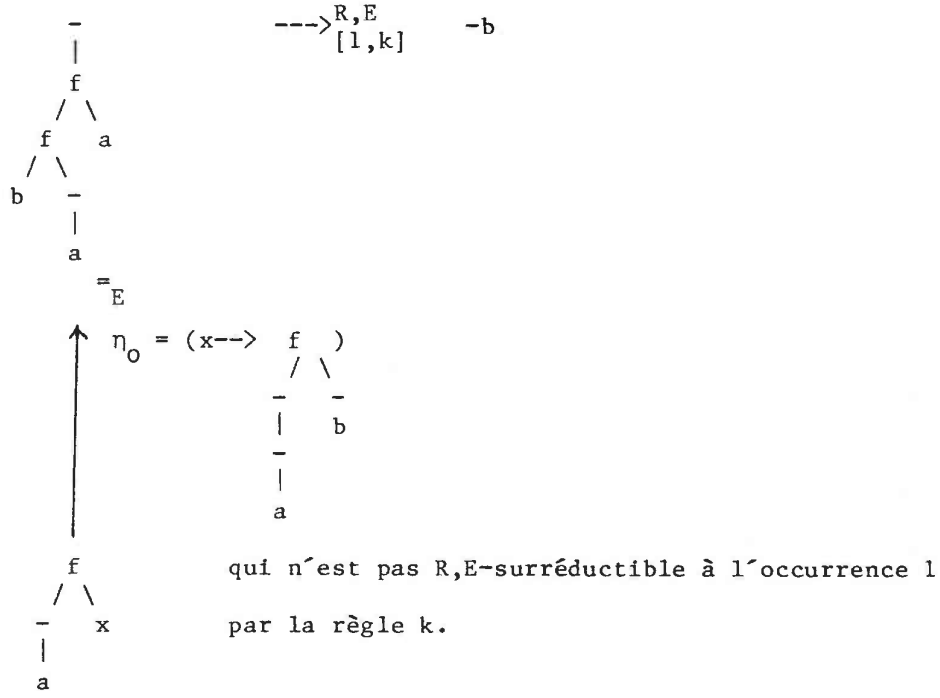
$$-f(x',y') = f(-y',-x').$$

Il est facile de se convaincre tout d'abord qu'il n'est plus possible d'avoir, avec les notations du théorème 4.1, $\eta_i(u_i) = v_i$ à chaque étape:



avec σ_0 = (x -> f).

Dʼautre part, en supposant quʼà chaque étape $\eta_i(u_i) =_E v_i$, on ne peut plus "calquer" les occurrences et les règles appliquées dans la R,E-réduction et dans la R,E-surréduction:



qui n'est pas R,E-surréductible à l'occurrence 1 par la règle k.

La condition supplémentaire à imposer est la propriété de E-commutation du système de réécriture R, que nous allons étudier maintenant. Elle va nous permettre d'utiliser une R,E-dérivation intermédiaire dans laquelle, à chaque étape, l'occurrence et la règle utilisées seront les mêmes dans la R,E-réduction et la R,E-surréduction.

4.2.2- LA PROPRIETE DE E-COMMUTATION

DEFINITION 4.6: Une relation ---> est E-commutante si et seulement si, pour tous termes t, t' et t_1 tels que $t' =_E t \longrightarrow t_1$, il existe t'_1 tel que $t' \longrightarrow t'_1$ et $t'_1 =_E t_1$. Cette propriété se visualise par le diagramme suivant:

$$\begin{array}{ccc} t & \longrightarrow & t_1 \\ =_E & & =_E \\ t' & \longrightarrow & t'_1 \end{array}$$

Avant d'utiliser cette propriété dans le cadre de la R,E-surréduction, nous allons nous intéresser à sa décidabilité.

Dans le cas où les axiomes de E permettent de définir une relation de réécriture $--->^E$, nous allons d'abord localiser cette propriété de la facon suivante:

DEFINITION 4.7: Une relation ---> est localement E-commutante si et seulement si pour tous termes t, t' et t_1 tels que $t' <--->^E t ---> t_1$, il existe un terme t'_1 tel que $t' ---> t'_1$ et $t'_1 =_E t_1$. Ce qui se visualise par le diagramme:

$$\begin{array}{ccc} t & ---> & t_1 \\ <--->^E & & =_E \\ t' & ---> & t'_1 \end{array}$$

PROPOSITION 4.2: Une relation ---> est E-commutante si et seulement si elle est localement E-commutante.

Preuve: la condition suffisante se prouve en notant que $=_E$ est $<-\underline{n}->^E$ et en faisant une récurrence sur n. []

Notre but étant maintenant d'étudier la E-commutation de $--->^{R,E}$, il faut tout d'abord remarquer que, dans la définition de la E-commutation de $--->^{R,E}$, on peut remplacer l'hypothèse $t--->^{R,E} t_1$ par $t--->^R t_1$ ou par $t--->^{R/E} t_1$ et obtenir des définitions équivalentes à la première.

D'autre part, il serait agréable de pouvoir caractériser cette propriété sur les règles de réécriture de R et les axiomes de E, en utilisant la notion de paires E-critiques de R/E. C'est ce que nous allons faire maintenant.

PROPOSITION 4.3: Soit E un ensemble d'axiomes g=d tels que g et d soient linéaires et que V(g)=V(d). Soit Γ un ensemble complet de paires E-critiques de R/E.

* Si la relation $\xrightarrow{R,E}$ est localement E-commutante, alors toute paire E-critique (u, v) de R/E dans Γ vérifie $v \xrightarrow{R,E} u' =_E u$

* Si toute paire E-critique (u,v) de Γ vérifie:

$v = \sigma(d) \xrightarrow{R,E} u'$ à une occurrence de d et $u' =_E u$,

alors la relation $\xrightarrow{R,E}$ est localement E-commutante.

Preuve: elle s'inspire de la preuve du théorème de Knuth et Bendix donnée par Huet [HUE,81].

* Supposons la relation $\xrightarrow{R,E}$ localement E-commutante. Soit alors (u,v) une paire E-critique de R/E dans Γ : il existe alors un axiome g=d dans EUE^{-1}, une règle $g' \rightarrow d'$ dans R, une occurrence m de g et une substitution σ tels que $u=\sigma(g[m\leftarrow d'])$ et $v=\sigma(d)$. Il est clair que le terme $\sigma(g)$ vérifie $\sigma(g) \xrightarrow{R,E} u$, puisque $\sigma(g)_{|m} =_E \sigma(g')$, et $\sigma(g) =_E v$. En appliquant l'hypothèse de locale E-commutation,

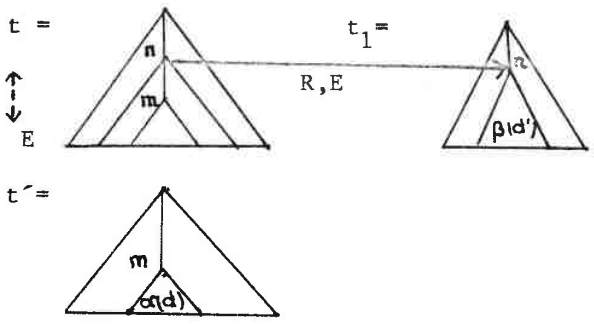
$$v \xrightarrow{R,E} u' =_E u.$$

* Supposons maintenant la propriété vraie sur les paires E-critiques de Γ et considérons des termes t, t' et t_1 vérifiant: $t \xrightarrow{R,E} t_1$ et $t \xleftrightarrow{E} t'$. Désignons par m, l'occurrence d'application dans t de l'axiome g=d, et par n, l'occurrence d'application dans t de la règle $g' \rightarrow d'$. Trois cas sont à distinguer:

** m et n sont des occurrences disjointes.

Dans ce cas, $\xrightarrow{R,E}$ et $\xleftarrow{E}$ commutent de manière évidente.

** n est préfixe de m ou m=n.



Posons $t_{|m} = \alpha(g)$ et $t_{|n} =_E \beta(g')$

On a $t'=t[m\leftarrow\alpha(d)]$ et $t_1=t[n\leftarrow\beta(d')]$

Il est clair que $t'_{|n} =_E t_{|n} =_E \beta(g')$ et donc

$t' \xrightarrow{R,E} t'_1 = t'[n<-\beta(d')] = t[n<-\beta(d')] = t_1$.

On a donc dans ce cas : $t \xrightarrow{R,E} t_1$, les deux R,E-réductions

$t \updownarrow_E t'$, $t' \xrightarrow{R,E} t_1$

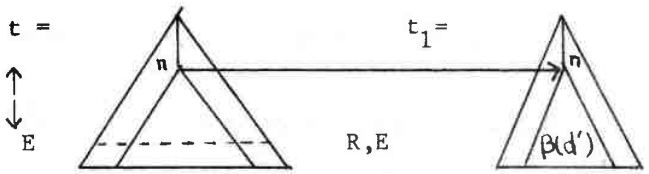
utilisant la même règle et le même E-filtre β .

** m est préfixe de n.

Pour simplifier les notations, nous supposerons que $m=\varepsilon$.

Deux sous-cas sont à nouveau à examiner:

*** n est une occurrence de O(g).

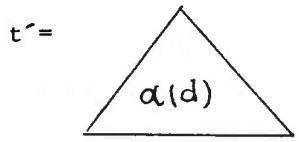


$t = \alpha(g)$

$t_{|n} =_E \beta(g')$

$t_1 = \alpha(g)[n<-\beta(d')]$

$t' = \alpha(d)$



α et β ayant des domaines disjoints, on montre facilement que la substitution $\alpha \cup \beta$ E-unifie $g_{|n}$ et g':

$\alpha \cup \beta(g_{|n}) = \alpha(g_{|n}) = \alpha(g)_{|n} =_E \beta(g') = \alpha \cup \beta(g')$.

Il existe donc une substitution σ appartenant à un ensemble complet de E-unificateurs de $g_{|m}$ et de g' en dehors de $V(g) \cup V(g')$ tel que $\sigma <_E \alpha \cup \beta\ [V(g) \cup V(g')]$, donc une substitution ρ vérifiant:

$\rho\sigma =_E \alpha \cup \beta\ [V(g) \cup V(g')]$.

$u = \sigma(g[n<-d'])$ et $v = \sigma(d)$ constituent alors une paire E-critique de R/E dans Γ vérifiant de plus $\rho(u) =_E t_1$ et $\rho(v) =_E t'$.

Par hypothèse sur les paires E-critiques:

$$
\begin{array}{ccc}
v = \sigma(d) & \xrightarrow{R,E} & u' =_E \sigma(g[n\leftarrow d']) = u \\
\rho \downarrow & & \downarrow \rho \\
\rho\,\sigma(d) & & \rho\,\sigma(g)[n\leftarrow \rho\,\sigma(d')] \\
=_E & & =_E \\
\alpha(d)=t' & & \alpha(g)[n\leftarrow\beta(d')] = t_1
\end{array}
$$

Désignons par n' l'occurrence d'application de la R,E-réduction dans $\sigma(d)$, par g"--->d" la règle appliquée et par μ un E-filtre de g" vers $\sigma(d)_{|n'}$; n' étant une occurrence de d par hypothèse,

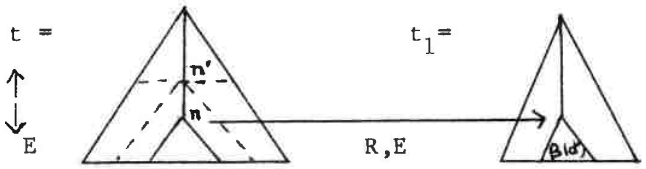
$\alpha(d)_{|n'} = \alpha(d_{|n'}) =_E \rho\,\sigma(d_{|n'}) =_E \rho\,\mu(g")$

donc $\alpha(d) \xrightarrow{R,E} \alpha(d)[n'\leftarrow\rho\,\mu(d")] =_E \rho(\sigma(d)[n'\leftarrow\mu(d")]) = \rho(u')$

Nous obtenons d'autre part:

$t_1 = \alpha(g)[n\leftarrow\beta(d')] =_E \rho\,\sigma(g)[n\leftarrow\rho\,\sigma(d')] = \rho(u) =_E \rho(u')$.

*** n n'est pas une occurrence de O(g).

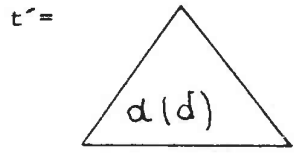


$t = \alpha(g)$

$t_{|n} =_E \beta(g')$

$t_1 = \alpha(g)[n\leftarrow\beta(d')]$

$t' = \alpha(d)$



Il existe donc une variable x à une occurrence n' dans g telle que $\alpha(x)$ contienne le sous-terme $t_{|n}$ à l'occurrence n".

Définissons la substitution μ sur V(g) par:

$\mu(x) = \alpha(x)[n"\leftarrow\beta(d')]$

$\mu(y) = \alpha(y)$ pour toute variable différente de x.

Puisque $\alpha(x) \xrightarrow{R,E} \mu(x)$ et que d est linéaire en x,

$t' = \alpha(d) \xrightarrow{R,E} \mu(d)$.

D'autre part, $\mu(g)=\alpha(g)[n'\leftarrow\mu(x)]=\alpha(g)[n'\leftarrow\alpha(x)[n"\leftarrow\beta(d')]]$

$=\alpha(g)[n\leftarrow\beta(d')] = t_1$

et $t_1 = \mu(g) \xleftrightarrow{E} \mu(d)$.

Remarquons ici encore que la R,E-réduction sur t' se fait avec la même

règle et le même E-filtre β que la R,E-réduction sur t.

Nous avons ainsi prouvé que la relation $--->^{R,E}$ est localement E-commutante. []

REMARQUE: Dans le cas où l'on utilise les E-paires critiques de R/E, il apparait clairement dans la preuve que σ (d) doit se R,E-réécrire à une occurrence appartenant à dom(d). Cette hypothèse est indispensable dans le cas général, mais peut-être est il possible de l'affaiblir pour certains axiomes particuliers. Cette question reste ouverte.

L'hypothèse de linéarité des axiomes est essentielle dans la dernière partie de la preuve et il est facile de construire un contre-exemple dans lequel la violation de cette condition empêche la E-commutation:

En prenant E réduit à l'axiome $f(x,f(y,x)) = f(x,f(x,y))$

et R réduit à la règle $f(z,a) ---> a$

il est facile de constater que, en appliquant l'axiome à l'occurrence ε et la règle à l'occurrence 22:

$f(f(u,a),f(b,f(u,a))) --->^{R,E} f(f(u,a),f(b,a))$

$=_E$

$f(f(u,a),f(f(u,a),b)) --->^{R,E} f(f(u,a),f(a,b))$

ces deux derniers termes n'étant pas E-égaux.

Cette condition sur E ne permet pas de considérer des axiomes tels qu'une loi d'idempotence pour un symbole de fonction f, $f(x,x) = x$, ou l'existence d'un inverse, $f(x,-x) = e$, mais de toute facon, quel que soit le système de règles R et les autres axiomes associés, ces deux axiomes conduisent à des relations $--->^{R/E}$ qui n'ont pas la propriété de terminaison finie et l'étude de la théorie ne pourra pas se faire en utilisant la relation $--->^{R,E}$ et la méthode développée par Peterson et Stickel [P&T,81].

4.2.3- RELATION ENTRE R,E-REDUCTION ET R,E-SURREDUCTION

Nous pouvons maintenant donner une généralisation du théorème 4.1.

THEOREME 4.4: Soit R un système de réécriture, E un ensemble d'axiomes tel qu'on connaisse un algorithme de E-unification complet. On suppose de plus que la relation $--->^{R,E}$ est E-commutante.

Soit u un terme, W un ensemble fini de variables contenant V(u), η une substitution R,E-normalisée avec $D(\eta)$ contenu dans V(u).

Alors pour toute R,E-dérivation (1) issue de $v =_E \eta(u)$, il existe une R,E-surdérivation associée (2) issue de u:

$$\begin{array}{ccccccc}
\eta(u) =_E v_0 & --->^{R,E}_{[m_0,k_0]} & v_1 & ---> & .... & --->^{R,E}_{[m_{n-1},k_{n-1}]} & v_n \quad (1) \\
\uparrow \eta_0 & & \uparrow =_E \eta_1 & & & & \uparrow =_E \eta_n \\
u=u_0 -V--->^{R,E}_{[m'_0,k'_0,\sigma_0]} & u_1 -V---> & .... & & -V--->^{R,E}_{[m'_{n-1},k'_{n-1},\sigma_{n-1}]} & u_n & (2)
\end{array}$$

et, pour tout i compris entre 0 et n, une substitution η_i et un ensemble de variables V_i tels que:

a) $D(\eta_i)$ est inclus dans V_i

b) η_i est R,E-normalisée

c) $\eta_{|W} =_E (\eta_i \theta_i)_{|W}$ et $I(\theta_i)$ est inclus dans V_i

où θ_i est définie récursivement par

$\theta_0 = Id$ et

$\theta_{i+1} = \sigma_i \theta_i$ pour i compris entre 0 et n-1.

d) $V(u_i)$ est inclus dans V_i et $\eta_i(u_i) =_E v_i$

Réciproquement, à chaque R,E-surdérivation (2) et à chaque substitution η telle que $\theta_n \leq_E \eta\ [W]$ est associée une R,E-dérivation (1).

Preuve: Une R,E-dérivation étant donnée, la preuve de l'existence d'une R,E-surdérivation vérifiant les conditions a,b,c,d se fait par

récurrence sur l'entier i:

* pour i=0, le résultat est clair en prenant $\eta_0 = \eta$

$I(\eta_0) = \emptyset$

$V_0 = WUD(\eta) = W$.

* supposons maintenant la R,E-surdérivation construite jusqu'au rang i et vérifiant a,b,c,d pour i.

Nous allons alors construire le diagramme suivant:

$$\begin{array}{ccccc} v_i & & \xrightarrow{R,E}_{[m_i,k_i]} & & v_{i+1} \\ =_E & & & & =_E \\ \eta_i(u_i) & & \xrightarrow{R,E}_{[m'_i,k'_i]} & & v'_{i+1} \\ \eta_i \uparrow & & & & \uparrow \eta_{i+1} \\ u_i & & \xrightarrow{R,E}_{[m'_i,k'_i,\sigma_i]} & & u_{i+1} \end{array}$$

** Montrons tout d'abord que u_i est R,E-surréductible: utilisant le fait que, par hypothèse de récurrence, $\eta_i(u_i) =_E v_i$ et que la relation $\xrightarrow{R,E}$ est E-commutante, on en déduit l'existence d'une occurrence m'_i de $dom(\eta_i(u_i))$, d'une règle k'_i et d'une substitution σ' telles que

$$\sigma'(g_{k'_i}) =_E \eta_i(u_i)|_{m'_i}$$

$$\eta_i(u_i) \xrightarrow{R,E}_{[m'_i,k'_i]} v'_{i+1} = \eta_i(u_i)[m'_i \leftarrow \sigma'(d_{k'_i})]$$

et l'on supposera que les variables de g_{k_i} ont été renommées de telle sorte que $D(\sigma')$ soit d'intersection vide avec V_i. La propriété de E-commutation permet en outre d'affirmer que

$$v'_{i+1} =_E v_{i+1}.$$

η_i étant R,E-normalisée par hypothèse de récurrence, m'_i appartient à $O(u_i)$ et donc:

$$(\eta_i(u_i))|_{m'_i} = \eta_i((u_i)|_{m'_i}) =_E \sigma'(g_{k_i}).$$

Puisque $D(\eta_i)$ est inclus dans V_i, η_i et σ' ont des domaines disjoints et la substitution $\eta_i U \sigma'$ est parfaitement définie. Elle E-unifie $(u_i)_{|m'_i}$ et $g_{k'_i}$ et il existe alors une substitution σ_i appartenant à un ensemble complet de E-unificateurs de $(u_i)_{|m'_i}$ et de $g_{k'_i}$ en dehors de $V_i UV(g_{k'_i})$ vérifiant

$$\sigma_i \leq_E \eta_i U \sigma' \qquad [V_i UV(g_{k'_i})]$$

On en déduit l'existence d'une substitution η' telle que

$$\eta' \sigma_i =_E \eta_i U \sigma' \qquad [V_i UV(g_{k'_i})]$$

et de la R,E-surréduction de u_i

$$u_i \xrightarrow[[m'_i, k'_i, \sigma_i]]{R,E} u_{i+1} \text{ avec } u_{i+1} = \sigma_i(u_i[m'_i \leftarrow g_{k'_i}])$$

** Définissons maintenant η_{i+1} et V_{i+1} en posant

$V_{i+1} = V_i \cup I(\sigma_i) \setminus D(\sigma_i)$ et $\eta_{i+1} = \eta'_{|V_{i+1}}$.

Remarquons que ces définitions impliquent:

d'une part que $\eta_i =_E (\eta_{i+1} \sigma_i)_{|V_i}$

puisque $\eta_i =_E (\eta' \sigma_i)_{|V_i}$ et que $I(\sigma_i)$ est inclus dans V_{i+1}

d'autre part que $V(u_{i+1})$ est inclus dans V_{i+1} si l'on suppose $V(u_i)$ inclus dans V_i, ce qui est vrai pour i=0.

** Vérifions alors la condition d) pour i+1, c'est-à-dire:

$$\eta_{i+1}(u_{i+1}) =_E v_{i+1}$$

Par la construction faite précédemment:

$$\eta_{i+1}(u_{i+1}) = \eta_{i+1}(\sigma_i(u_i[m'_i \leftarrow d_{k'_i}]))$$
$$= (\eta_{i+1} \sigma_i(u_i))[m'_i \leftarrow \eta_{i+1} \sigma_i(d_{k'_i})]$$

Or $\eta_{i+1} \sigma_i(d_{k'_i}) = \eta' \sigma_i(d_{k'_i})$ puisque $\sigma_i(d_{k'_i})$ est un sous-terme de u_{i+1} et est donc E-égal à $\sigma'(d_{k'_i})$.

Donc $\eta_{i+1}(u_{i+1}) =_E (\eta_{i+1} \sigma_i(u_i))[m'_i \leftarrow \sigma'(d_{k'_i})]$

$$=_E (\eta_i(u_i))[m'_i \leftarrow \sigma'(d_{k'_i})] = v'_{i+1}$$

car $\eta_i =_E (\eta_{i+1} \sigma_i)_{|V_i}$.

D'où $\eta_{i+1}(u_{i+1}) =_E v'_{i+1} =_E v_{i+1}$.

** Vérifions enfin les conditions a,b,c pour i+1:

a) $D(\eta_{i+1})$ est inclus dans V_{i+1} par définition de η_{i+1}.

b) η_{i+1} est R,E-normalisée car

- si x est une variable de $V_i \backslash D(\sigma_i)$ alors

  $\eta_{i+1}\ \sigma_i(x) = \eta_{i+1}(x) =_E \eta_i(x)$ et comme $\eta_i(x)$ est R,E-irréductible par hypothèse, $\eta_{i+1}(x)$ l'est également, la relation $--->^{R,E}$ étant E-uniforme.

- si x est une variable de $I(\sigma_i)$, il existe une variable y de $D(\sigma_i) \cap V_i$ telle que $\sigma_i(y)=x$.

  Donc $\eta_{i+1}(x) = \eta_{i+1}\ \sigma_i(y) =_E \eta_i(y)$ et l'on conclut comme dans le cas précédent.

c) $\eta_{|W} =_E (\eta_{i+1}\ \theta_{i+1})_{|W}$ car

$\eta_{|W} =_E (\eta_i\ \theta_i)_{|W}$ par hypothèse de récurrence

$=_E ((\eta_{i+1}\ \sigma_i)_{|V_i}\ \theta_i)_{|W}$.

Il est facile de prouver par récurrence que $I(\theta_i)$ est inclus dans V_i et on peut alors écrire

$(\eta_i\ \theta_i)_{|W} =_E (\eta_{i+1}\ \sigma_i\ \theta_i)_{|W}$

$=_E (\eta_{i+1}\ \theta_{i+1})_{|W}$ par hypothèse de récurrence.

Nous avons ainsi terminé la preuve de la première partie du théorème.

** Prouvons maintenant qu'à chaque R,E-surdérivation issue de u, on peut associer une R,E-dérivation issue d'un terme E-égal à $\eta(u)$, η étant une substitution vérifiant $\theta_n \leq_E \eta\ [W]$.

Soit ρ une substitution telle que $(\rho\ \theta_n)_{|W} =_E \eta_{|W}$.

Posons pour tout i compris entre 1 et n-1, $\eta_i = \eta_{i+1}\ \sigma_i$

$$\eta_n = \rho$$

$$v_i = \eta_i(u_i)$$

Nous allons construire par récurrence une R,E-dérivation

$$\eta(u) =_E v_0 \xrightarrow[[m_0,k_0]]{R,E} v_1 \longrightarrow \dots \xrightarrow[[m_{n-1},k_{n-1}]]{R,E} v_n$$

Pour i quelconque compris entre 0 et n-1,

$$\begin{array}{ccc} u_i & \xrightarrow[[m_i,k_i,\ \sigma_i]]{R,E} & u_{i+1} \\ \downarrow \eta_i & & \downarrow \eta_{i+1} \\ v_i & & v_{i+1} \end{array}$$

* Montrons que v_i est R,E-réductible en v_{i+1} :

$v_{i|m_i} = (\eta_i(u_i))_{|m_i} = \eta_{i+1}\ \sigma_i((u_i)_{|m_i})$ par définition des η_i et de m_i.

Comme $\sigma_i((u_i)_{|m_i}) =_E \sigma_i(g_{k_i})$, $(v_i)_{|m_i} =_E \eta_{i+1}\ \sigma_i(g_{k_i})$

Donc $v_i \xrightarrow[[m_i,k_i]]{R,E} v_i[m_i <- \eta_{i+1}\ \sigma_i(g_{k_i})]$

$= \eta_i(u_i)[m_i <- \eta_{i+1}\ \sigma_i(g_{k_i})]$

$= \eta_{i+1}\ \sigma_i(u_i)[m_i <- \eta_{i+1}\ \sigma_i(g_{k_i})]$

$= \eta_{i+1}(\sigma_i(u_i[m_i <- \eta_{i+1}\ \sigma_i(g_{k_i})]))$

$= \eta_{i+1}(u_{i+1}) = v_{i+1}$ par construction.

* Montrons enfin que $v_0 =_E \eta(u)$:

$v_0 = \eta_0(u_0) = \eta_0(u) = \eta_1\ \sigma_0(u) = \dots$

$= \eta_n\ \sigma_{n-1} \dots \sigma_0(u) = \eta_n\ \theta_n(u) = \rho\ \theta_n(u)$

$=_E \eta(u)$ car V(u) est inclus dans W.

Ceci termine donc la preuve de cette proposition, mais nous allons la compléter par la remarque suivante, utile par la suite: dans la démonstration de la réciproque, nous avons construit la R,E-dérivation de telle sorte que $\eta(u) =_E v_0$

et pour tout i compris entre 1 et n, $\eta_i(u_i) = v_i$.

Mais il est possible de construire une R,E-dérivation issue de $\eta(u)$ cette fois-ci et vérifiant $\eta_i(u_i) =_E v_i$ pour tout i compris entre 1 et n. En effet, la relation $\xrightarrow{R,E}$ étant E-commutante,

$$v_0 \xrightarrow{R,E}_{[m_0,k_0]} v_1$$
$$=_E$$
$$\eta(u)$$

implique que $\eta(u)$ est R,E-réductible en $v'_1 =_E v_1 = \eta_1(u_1)$ et en réitérant ce raisonnement, on construit la R,E-dérivation

$$\eta(u) \xrightarrow{R,E}_{[m'_0,k'_0]} v'_1 \longrightarrow \ldots \xrightarrow{R,E}_{[m'_{n-1},k'_{n-1}]} v'_n$$
$$=_E \qquad\qquad\qquad\qquad\qquad\qquad =_E$$
$$\eta_1(u_1) \qquad\qquad\qquad\qquad\qquad\qquad \eta_n(u_n)$$

Ceci nous permet d'affirmer que le théorème reste vrai si l'on considère des R,E-dérivations issues de $v_0 = \eta(u)$. []

4.2.4- A-UNIFICATION ET R,E-SURREDUCTION

Comme précédemment, nous allons établir deux lemmes permettant de préciser le lien entre unification et R,E-surréduction; cela nous permettra de donner un algorithme de A-unification avec A=RUE.

De même que dans le cas d'une théorie où le système de réécriture associé est canonique, on va R,E-surréduire les deux termes t et t' dont on souhaite trouver un ensemble complet de A-unificateurs, simultanément et jusqu'à ce que les termes t_n et t'_n obtenus soient E-unifiables.

Pour itérer ces surréductions en parallèle sur les deux termes, nous R,E-surréduisons le terme *(t,t').

Dans tout ce paragraphe nous garderons les mêmes notations que dans ce qui précède, en particulier celles du théorème 4.4.

Nous commencons par montrer comment, en combinant la E-unification et la R,E-surréduction, on obtient bien un A-unificateur de t et t'.

LEMME 4.5: Considérons la R,E-surdérivation issue de $*(t,t')$

$$u = *(t,t') = u_0 \text{ -V---}>^{R,E} u_1 = *(t_1,t'_1) \text{ -V---}>^{R,E} \ \ldots \ \text{ -V---}>^{R,E} u_n = *(t_n,t'_n)$$

telle que t_n et t'_n soient E-unifiables par μ;

alors $\mu\theta_n$ est un A-unificateur de t et t', où θ_n est la composition des substitutions surréductrices faites au cours de la R,E-surdérivation.

Preuve: On utilise la réciproque du théorème 4.4 avec $\mu = \theta_n$.

On associe donc à la R,E-surdérivation donnée la R,E-dérivation:

$$\theta_n(u) = v_0 \text{ --->}^{R,E} v_1 \text{ --->}^{R,E} v_2 \text{ ---> } \ldots \text{ --->}^{R,E} v_n = *(s_n,s'_n)$$

et donc $\theta_n(t) \text{---}\overset{*}{>}^{R,E} s_n$ et $\theta_n(t') \text{---}\overset{*}{>}^{R,E} s'_n$

mais ici nous avons η_n=Id et par conséquent

$$s_n =_E t_n \text{ et } s'_n =_E t'_n$$

or $\mu\theta_n(t) =_A \mu(t_n) =_E \mu(t'_n) =_A \mu\theta_n(t')$

donc $\mu\theta_n(t) =_A \mu\theta_n(t')$. []

Un second lemme va maintenant prouver que tout A-unificateur peut être atteint par ce procédé.

LEMME 4.6: Soient t et t' deux termes A-unifiables par ρ et W un ensemble fini de variables contenant V(t)UV(t').

Alors il existe une R,E-surdérivation issue de $u=*(t,t')$

$$u = *(t,t') = u_0 \text{ -V---}>^{R,E} u_1 \text{ -V---> } \ldots \text{ -V---}>^{R,E} u_n = *(t_n,t'_n)$$

telle que t_n et t'_n soient E-unifiables. De plus:

-- d'une part, en posant W_n=V(t_n)UV(t'_n), il existe μ élément de

Σ ensemble complet de E-unificateurs de t_n et t'_n en dehors de WUW_n tel que

$$\mu\theta_n \leq_A \rho \quad [W]$$

-- d'autre part, on peut se restreindre aux R,E-surdérivations telles que pour tout entier i, $0 \leq i \leq n$, $(\theta_i)_{|W}$ est R,E-normalisée.

Preuve: Nous avons $\rho(t) =_A \rho(t')$ et en prenant $\eta = R,E\text{-}FN(\rho)$ (i.e. on R,E-normalise ρ) on a $\eta(t) =_A \eta(t')$.

Soit s et s' les R,E-formes normales de $\eta(t)$ et $\eta(t')$ respectivement.

Il existe donc une R,E-dérivation:

$$\eta(u) = *(\eta(t),\eta(t')) = v_0 \ --->^{R,E} \ \ldots \ --->^{R,E} \ v_n = *(s,s')$$

La R,E-surdérivation associée à cette R,E-dérivation par le théorème 4.4 est telle que:

$$\eta_n(u_n) = *(\eta_n(t_n),\eta_n(t'_n)) =_E v_n = *(s,s')$$

Or $s =_E s'$ donc $\eta_n(t_n) =_E \eta_n(t'_n)$ et par conséquent

η_n est un E-unificateur de t_n et t'_n .

Σ étant un ensemble complet de E-unificateurs de t_n et t'_n en dehors de WUW_n il existe μ dans Σ tel que

$$\mu \leq_E \eta_n \ [WUW_n]$$

et il existe ρ telle que

$$\rho\mu' =_E \eta_n \ [WUW_n]$$

ce qui équivaut à

$$(\rho\ \mu)_{|WUW_n} =_E (\eta_n)_{|WUW_n}$$

D'où $((\rho\ \mu)_{|WUW_n})((\theta_n)_{|W}) =_E ((\eta_n)_{|WUW_n})((\theta_n)_{|W})$

mais par construction $I(\theta_n)$ est inclus dans W_n, donc

$$(\rho\ \mu\ \theta_n)_{|W} =_E (\eta_n\ \theta_n)_{|W}$$

Donc $\mu\ \theta_n \leq_E \eta_n\ \theta_n\ [W]$ et comme $(\eta_n\ \theta_n)_{|W} = \eta_{|W} =_A \rho_{|W}$

on a $$\mu\ \theta_n \leq_A \rho \quad [W]$$

De plus les $(\theta_i)_{|W}$ sont R,E-normalisées puisque

$(\eta_n\ \theta_n)_{|W} =_E \eta_{|W}$ et que η est R,E-normalisée. []

Nous pouvons donc maintenant énoncer le résultat établissant le lien entre la A-unification et la R,E-surréduction.

THEOREME 4.5: Soit A une théorie équationnelle définie par A=RUE où

. R est un système de réécriture E-canonique tel que $--->^{R,E}$ soit E-commutante

. E est un ensemble d'équations pour lesquelles un algorithme de E-unification complet est connu.

Soient t et t' deux termes, u=*(t,t') où * est un "nouveau" symbole et W un ensemble fini de variables contenant V(u). Soit Σ l'ensemble des substitutions σ vérifiant: il existe une ,E-surdérivation

$$u = *(t,t') \ -V--->^{R,E} \ u_1 \ -V--->^{R,E} \ \ldots. \ -V--->^{R,E} \ u_n = *(t_n,t'_n)$$

telle que les termes t_n et t'_n soient E-unifiables,

θ_n soit R,E-normalisée

et $\sigma = \mu \theta_n$

où μ est un élément d'un ensemble complet de E-unificateurs de t_n et t'_n en dehors de WUW_n, W_n étant un ensemble fini de variables contenant $V(u_n)$. Alors Σ est un ensemble complet de A-unificateurs de t et t' en dehors de W.

La preuve résulte des deux lemmes précédents. []

L'algorithme découlant de ce résultat consiste donc en l'exploration systématique de l'arbre des R,E-surdérivations issues de u=*(t,t'). Un tel processus ne terminera pas si, par exemple, les ensembles de E-unificateurs peuvent être infinis ou si une R,E-surdérivation ne termine pas. Nous allons nous attacher maintenant à diminuer l'ensemble des R,E-surréductions "intéressantes", c'est-à-dire nécessaires à la construction d'un ensemble complet de A-unificateurs.

4.2.5- LA R,E-SURREDUCTION BASIQUE

Dans cette partie, nous allons donner une généralisation de la notion de surréduction basique vue précédemment. Le théorème justifiant l'utilisation de la surdérivation basique ne s'étend pas à la R,E-surréduction, même si l'on suppose

que R est E-commutante. Il nous faudra renforcer cette hypothèse, de facon à ce que, si $u =_E v \;\;\rightarrow^{R,E}\;\; w$ et que la R,E-réécriture se fait par une substitution R,E-normalisée, alors la R,E-réécriture issue de u se fasse également par une substitution R,E-normalisée. Sous cette hypothèse, que nous appellerons stricte E-commutation de $\rightarrow^{R,E}$, les résultats de la partie précédente s'étendent sans peine.

La définition d'une R,E-dérivation basée sur un ensemble d'occurrences U est exactement semblable à celle donnée plus haut, d'une dérivation basée sur U. Montrons tout d'abord que la classe des R,E-dérivations basées contient les dérivations suivant une stratégie de l'intérieur vers l'extérieur.

DEFINITION 4.8: Une R,E-dérivation suit une stratégie de l'intérieur vers l'extérieur si et seulement si, pour toute étape $u_i \;\rightarrow^{R,E}_{[m_i,k_i]}\; u_{i+1}$, la substitution σ telle que $u_{i+1} = u_i[m_i <- \sigma(d_{k_i})]$ est R,E-normalisée.

LEMME 4.7: Soit $v = \eta(u)$, avec η R,E-normalisée. Toute R,E-dérivation issue de v suivant une stratégie de l'intérieur vers l'extérieur est basée sur O(u).

Preuve: Les substitutions étant toutes R,E-normalisées, une R,E-dérivation est basée sur O(u) par définition. []

LEMME 4.8: Soit $v = \eta(u)$ où η est une substitution R,E-normalisée. Il existe une R,E-dérivation issue de v vers la forme normale de v qui est basée sur O(u).

Preuve: Le système de réécriture étant E-canonique, il existe une R,E-dérivation issue de v vers la forme normale de v, qui suit une stratégie de l'intérieur vers l'extérieur. Il suffit d'appliquer alors le lemme précédent. []

REMARQUE: La terminaison finie du système de réécriture est ici superflue: il suffit qu'il existe des formes normales et de bonnes stratégies de calcul. Par contre, la E-confluence est indispensable. Le processus de R,E-surréduction basique peut donc être utilisé avec des systèmes de réécriture non canoniques mais ayant de bonnes propriétés.

DEFINITION 4.9: Une R,E-surdérivation est dite basique si et seulement si elle est basée sur O(u).

Cette définition est tout-à-fait identique à celle donnée dans le cas des surdérivations. Mais l'analogie s'arrête là, car on ne peut plus, comme dans le cas de la surréduction, calquer les occurrences d'application des R,E-surréductions sur celles des R,E-réductions dans une R,E-surdérivation et une R,E-dérivation associées , comme le prouve le théorème 4.4. Pour étendre les résultats, il nous faut faire une hypothèse plus forte que la E-commutation, que nous appellerons propriété de stricte E-commutation.

DEFINITION 4.10: La relation $--->^{R,E}$ sera dite strictement E-commutante si et seulement si pour tous termes u, v, w tels que

$$u =_E v \text{ et } v --->^{R,E}_{[m,k,\sigma]} w \qquad \text{avec } \sigma \text{ R,E-normalisée,}$$

il existe un terme u' tel que

$$u' =_E w \text{ et } u --->^{R,E}_{[m',k',\sigma']} u' \text{ avec } \sigma' \text{ R,E-normalisée.}$$

Le théorème 4.4 permet, une R,E-dérivation étant donnée, de trouver une une R,E-surdérivation et une R,E-dérivation telles que l'on ait le schéma:

$$\begin{array}{ccccccc}
v_0 & --->^{R,E}_{[m_0,k_0]} & v_1 & ---> \cdots & v_{n-1} & --->^{R,E}_{[m_{n-1},k_{n-1}]} & v_n \qquad (1)\\
{\scriptstyle =_E} & & {\scriptstyle =_E} & & {\scriptstyle =_E} & & {\scriptstyle =_E}\\
v'_0 & --->^{R,E}_{[m'_0,k'_0]} & v'_1 & ---> \cdots & v'_{n-1} & --->^{R,E}_{[m'_{n-1},k'_{n-1}]} & v'_n \qquad (2)\\
\uparrow{\scriptstyle \eta_0} & & \uparrow{\scriptstyle \eta_1} & & \uparrow{\scriptstyle \eta_{n-1}} & & \uparrow{\scriptstyle \eta_n}\\
u_0 & -V-->^{R,E}_{[m'_0,k'_0,\sigma_0]} & u_1 & -V--> \cdots & u_{n-1} & -V-->^{R,E}_{[m'_{n-1},k'_{n-1},\sigma_{n-1}]} & u_n \qquad (3)
\end{array}$$

Sous l'hypothèse de stricte E-commutation de la relation $--->^{R,E}$, si la R,E-réduction (1) se fait de l'intérieur vers l'extérieur, il en sera de même de la R,E-réduction (2). Par conséquent, puisque η_0 est R,E-normalisée et que les occurrences d'applications des règles sont les mêmes dans (2) et (3), d'une part la R,E-dérivation (2) est basée sur $O(u_0)$, d'autre part la R,E-surdérivation (3) est également basée sur $O(u_0)$. Cela va nous permettre de donner la spécialisation suivante du lemme 4.6.

LEMME 4.9: t et t' étant deux termes A-unifiables par la substitution ρ et W étant un ensemble fini de variables contenant V(t)UV(t'), il existe une R,E-surdérivation basique issue de u=*(t,t')

$$u = *(t,t') = u_0 \text{ -V---}>^{R,E} u_1 \quad \ldots \quad u_{n-1} \text{ -V---}>^{R,E} u_n = *(t_n,t'_n)$$

telle que t_n et t'_n soient E-unifiables. Soit μ un élément d'un ensemble complet de E-unificateurs de t_n et t'_n, nous avons alors:

$$\mu\, \theta_n \leq_E \rho \quad [W]$$

On peut de plus se restreindre aux R,E-surdérivations telles que pour tout entier i compris entre 0 et n, on ait $(\theta_i)_{|W}$ R,E-normalisée.

Preuve: Dans la preuve du lemme 4.6, la R,E-surdérivation utilisée est celle allant de η (u) à la forme normale de ce terme. On peut donc considérer une R,E-dérivation suivant une stratégie de l'intérieur vers l'extérieur. Le lemme 4.7 et la remarque ci-dessus permettent de conclure. []

On peut donc en déduire une spécialisation du théorème 4.5 aux R,E-surdérivations basiques:

THEOREME 4.6: Les conclusions du théorème 4.5 restent vraies si on considère uniquement les R,E-surdérivations basiques.

L'algorithme de A-unification issu de ce théorème termine si, par exemple, on peut montrer que toute R,E-surréduction diminue strictement le nombre d'occurrences basiques, c'est-à-dire le cardinal de l'ensemble d'occurrences U_i décrit plus haut, et que d'autre part, l'algorithme de E-unification est fini. On peut trouver une condition plus générale permettant d'affirmer que toute R,E-surdérivation est de longueur finie, puisque nous allons voir que la condition suffisante de terminaison donnée dans la cas de la surréduction s'applique encore au cas de la R,E-surréduction.

PROPOSITION 4.4: Soit R un système E-canonique de réécriture de termes tel que toute R,E-surdérivation basique issue d'un quelconque des membres droits de règles de R termine. Alors toute R,E-surdérivation basique issue d'un terme quelconque termine.

Preuve: Elle est tout-à-fait semblable à celle que donne J.M. Hullot [HUL,80] dans le cas d'un système de réécriture canonique.
L'idée de base de la preuve est la suivante: à chaque étape de la R,E-surdérivation, soit l'occurrence provient de O(u) et une telle occurrence ne peut être utilisée qu'une seule fois, soit cette étape de R,E-surdérivation "fait partie" d'une R,E-surdérivation issue d'un membre gauche de règle. []

COROLLAIRE 4.1: Si l'algorithme de E-unification est fini, sous les hypothèses de la proposition précédente, le processus décrit dans le théorème 4.6 permet de construire un algorithme de A-unification fini et complet.

Preuve: Elle découle des résultats précédents. []

CONCLUSION

Le résultat essentiel de ce chapitre est constitué par le théorème 4.5. En effet il ouvre de nouvelles perspectives dans la construction des algorithmes de A-unification, où A est décomposable en deux ensembles d'axiomes R et E tels qu'il existe un algorithme de E-unification complet et que l'on puisse orienter R, de facon à obtenir un système de réécriture qui soit E-canonique. La connaissance de l'algorithme de E-unification permet alors de construire (sous de bonnes hypothèses) un algorithme de A-unification. Mais le procédé est en quelque sorte incrémental puisque l'on peut alors le réitérer, en se donnant une nouvelle théorie A' contenant A, dans laquelle le rôle du E précédent sera joué par A. Il faudra alors vérifier, par exemple en utilisant les travaux de Peterson et Stickel [P&S,81] que le système constitué des axiomes de A'\A est E-confluent et E-noethérien.

On peut également voir ce mécanisme comme un algorithme de complétion incrémental, puisque la donnée de E et de son algorithme de E-unification complet permet de compléter R, en utilisant les résultats de Peterson et Stickel. Puis un algorithme de A-unification complet étant fourni par le processus de R,E-surréduction, on peut réitérer en ajoutant de nouvelles règles.

Nous montrons, dans un chapitre ultérieur, que ces résultats s'appliquent avec succès dans les arbres signés, et nous pensons qu'ils devraient permettre d'étudier de nouvelles théories équationnelles.

CHAPITRE CINQ

CONSTRUCTION D'UN ALGORITHME DE E-UNIFICATION DANS LES ARBRES SIGNES:

Une extension de l'algorithme de Martelli et Montanari

INTRODUCTION

Nous avons vu dans le chapitre 1 que, parmi les méthodes permettant de traiter séparément un ensemble particulier d'axiomes d'une théorie équationnelle, seule celle due à Peterson et Stickel permet de considérer des règles dont les membres gauches ne sont pas nécessairement linéaires.

Nous avons étudié dans le chapitre précédent une méthode permettant de calculer, sous de "bonnes" hypothèses un ensemble complet de A-solutions d'une équation: la R,E-surréduction. Nous prendrons ici $E = (A_1) \cup (A_2)$ (cf chapitre 2). Afin d'appliquer ces résultats à la théorie des algèbres signées et en particulier pour donner une méthode de résolution d'équations dans AS, il nous faut préalablement connaitre un algorithme de E-unification complet ainsi qu'un algorithme de E-filtrage. Ce chapitre est consacré à la résolution de ces deux problèmes.

Le premier, trouver un algorithme de E-filtrage, qui plus est complet et minimal, est simple. Le second est plus compliqué car, outre la difficulté de travailler dans la théorie équationnelle, nous verrons que les ensembles complets minimaux de E-unificateurs de deux termes dans AS peuvent être infinis.

Une autre particularité de cette recherche d'un algorithme de E-unification complet dans AS est la méthode utilisée. En effet nous avons été amenés à généraliser l'algorithme d'unification de Martelli et Montanari [M&M,79], à notre théorie équationnelle.

Rappelons brièvement que Martelli et Montanari ont décrit un algorithme d'unification dans l'ensemble des termes dont la conception est basée sur les deux principes suivants:

* D'une part la décomposition d'un système d'équations S en un système S' équivalent (i.e. ayant même ensemble de solutions) mais plus simple, au sens où les équations de S' sont plus faciles à résoudre que celles de S.

* D'autre part la forme des solutions des équations simplifiées n' est demandée qu'à la fin.

De facon plus précise, rappelons la démarche utilisée et décrite en détail dans [M&M,79].

-Tout d'abord on introduit la notion de multiéquation e comme étant la donnée d'un ensemble de variables V(e) et d'un multiensemble de termes (définition 1.34) note T(e). Initialement, si l'équation à résoudre est (t=t'), on prendra e=(t=t'). On notera e=(V(e)=T(e)); par exemple e_1=(x=y=f(a,z)=f(a,f(a,b))) où a et b sont des constantes, f un symbole d'arité 2 et x, y, z des variables.

-Pour se ramener à un système d'équations plus simple à résoudre que le système initial (t=t'), on décompose le système S de multiéquations en remarquant que pour une multiéquation e de S, tous les termes de T(e) doivent avoir un squelette (on pourrait dire une structure) commun et que le système de multiéquations aux différences issue des "différences" des termes de T(e) doit également être résolu.

Par exemple pour l'équation e_1 donnée plus haut, le squelette de $T(e_1)$ est f(a,z) et le système de multiéquations aux différences est réduit à (z=f(a,b)).

On peut donner une définition formelle du squelette sq(M) et du système de multiéquations aux différences EqD(M) d'un multiensemble de termes M de la facon suivante:

SI il existe dans M un terme réduit à une variable x

ALORS sq(M) = x et EqD(M) <- EqD(M) U (la multiéquation issue de M)

SINON SI tous les symboles de tête des termes de M sont égaux à f

ALORS SI f est un symbole de constante

ALORS sq(M) = f

SINON les p termes de M sont tous de la forme $f(t_1^j,\ldots,t_n^j)$ pour j compris entre 1 et p. Pour i compris entre 1 et n, soit M_i le multiensemble des t_i^j.

SI pour tout i compris entre 1 et n, $sq(M_i)$ existe

ALORS $sq(M) = f(sq(M_1,\ldots,M_n))$

$EqD(M) = U_{i=1,\ldots,n} EqD(M_i)$

SINON échec

FSI

FSI

SINON échec

FSI

FSI

-Il est alors nécessaire, pour construire une congruence d'unification cohérente, de regrouper toutes les multiéquations qui ont des ensembles de variables non disjoints. On définit donc la fusion de deux multiéquations e et e' telles que $V(e) \cap V(e') \neq \emptyset$ par la multiéquation

$$e'' = (V(e) \cup V(e') = T(e) \cup T(e'))$$

-L'algorithme consiste alors à itérer les phases de décomposition et de fusion. Mais il faut encore tenir compte de l'apparition de cycles dans la substitution solution. Ici encore une solution originale et élégante donnée par A.Martelli et U.Montanari consiste à tester au cours des phases de décomposition-fusion l'existence de tels cycles, à l'aide d'une relation

d'ordre notée $<$ sur les multiéquations, introduite à l'origine par G. Huet [HUE,76] pour des équations. On pose

$e_i < e_j$ si et seulement si,

il existe une variable de $V(e_i)$ qui apparait dans un des termes de $T(e_j)$.

On prouve alors que la fermeture transitive $\overset{+}{<}$ de $<$ est un ordre partiel sur les multiéquations de S, si le système S a une solution.

-Afin de faciliter l'expression de l'algorithme, on structure le multiensemble des équations S en un couple (T,R), où T est une suite de multiéquations telle que:

- $TUR = S$ et $T \cap R = \emptyset$
- $\forall\, e \in T$, $\#V(e) = 1$
- $\forall\, e_i, e_j \in T$, $i<j \Rightarrow e_i \not< e_j$

T constitue en quelque sorte l'ensemble des multiéquations dont on connait la solution. Si $R = \emptyset$, T détermine de facon évidente une solution du système S.

-En définitive, on obtient l'algorithme d'unification de deux termes suivant:

UNIFICATION

REPETER * Sélectionner une multiéquation de e de R maximale pour $\overset{+}{<}$. Si une telle équation n'existe pas, alors échec (il y a un cycle).

* SI $T(e) = \emptyset$

ALORS transférer e de R à la fin de T

SINON calculer le squelette et le système de multiéquations aux différences de T(e)

SI le squelette n'existe pas ALORS échec (par collision de symboles de fonctions)

SINON transférer la multiéquation $V(e) = sq(T(e))$ de R à T et fusionner $R \cup EqD(T(e))$

FSI

FSI

JUSQU'A $R = \emptyset$

arrêt avec succès.

Nous allons donc généraliser cette démarche au cas de notre théorie équationnelle. Mais comme nous le verrons, cette généralisation s'étend certainement à d'autres théories.

Le fil d'Ariane de ce chapitre est donc de parvenir à appliquer une démarche analogue à celle de Martelli et Montanari, à une théorie équationnelle RUE telle qu'il existe un système de réécriture confluent et à terminaison finie associé à E. Nous allons donc tout d'abord montrer qu'il est nécessaire et suffisant de savoir résoudre des équations dont les termes sont en E-forme normale. Puis nous donnerons des ensembles complets de E-unificateurs des équations $(x = t)$ avec $x \notin V(t)$ et $(x = -x)$. Enfin nous développerons un formalisme fortement inspiré de celui rappelé ci-dessus, pour donner un algorithme complet minimal de E-unification.

5.1- UN ALGORITHME DE E-FILTRAGE DANS AS

Rappelons que, dans toute la suite, E désigne l'ensemble des axiomes de (A1)U(A2).

Il est possible de déduire l'existence d'un algorithme de E-filtrage à partir de l'existence d'un algorithme de E-unification, mais nous préférons le construire explicitement, d'autant plus qu'il est particulièrement simple dans ce cas.

L'orientation des axiomes de E de gauche à droite permet de construire un système de réécriture confluent et noethérien, comme nous l'avons vu au chapitre précédent. Ce fait va permettre de raisonner sur des termes en E-forme normale (ou pré-forme normale notée pfn), en s'appuyant sur le résultat suivant:

LEMME 5.1: Pour tous termes t et t' de AS et toute substitution σ définie sur l'ensemble V(t) des variables de t, σ est un E-filtre de t vers t' si et seulement si σ est un E-filtre de pfn(t) vers pfn(t').

Preuve: $t =_E pfn(t)$ et $t' =_E pfn(t')$ impliquent:
pour toute substitution σ, $\sigma(t) =_E t' \iff \sigma(pfn(t)) =_E pfn(t')$.[]

Sur des termes en E-forme normale, il est alors très simple de détecter les cas d'échec du filtrage:

LEMME 5.2 : Soient t et t' deux termes de AS en E-forme normale.
Si $t=f(t_1,\ldots,t_n)$ avec f différent du symbole - et $t'=g(t'_1,\ldots,t'_m)$, alors

* si f et g sont distincts, il n'existe pas de E-filtre de t vers t'
* sinon tout E-filtre de t vers t' est un E-filtre de t_1 vers $t'_1,\ldots,$ et de t_n vers t'_n.

Preuve: elle est évidente en remarquant que si σ est un E-filtre de t vers t', $\sigma(t) = f(\sigma(t_1),\ldots,\sigma(t_n)) =_E g(t'_1,\ldots,t'_m)$.[]

Dès lors, un algorithme de descente récursive dans les deux termes permet de tester l'existence d'un E-filtre. On s'arrête en arrivant sur une variable ou une variable précédée du signe - dans t.

LEMME 5.3 : Si x est une variable, il existe un ensemble minimal complet de E-filtres de x vers t ou de -x vers t réduit à un seul élément.

Preuve: cet ensemble est réduit à la substitution (x-->t) dans le premier cas, et à la substitution (x-->m(t)) dans le second. La vérification de la complétude et de la minimalité sont triviales. []

Un ensemble minimal complet de E-filtres d'un terme t vers un terme t' est ainsi toujours réduit à un unique élément σ qui peut être choisi E-normalisé, c'est-à-dire tel que pour tout x dans son domaine $D(\sigma)$, $\sigma(x)$ est en E-forme normale.

PROPOSITION 5.1: L'algorithme suivant calcule l'unique élément d'un ensemble minimal complet de E-filtres d'un terme t vers un terme t'.
V=V(t) ; pour tout x dans V faire $\sigma(x)=x$.
t:=pfn(t) t':=pfn(t')

E-FILTRE (t,t´)

CAS * t=x, x appartient à V et σ(x)=x ALORS σ(x):=t

* t=-x, x appartient à V et σ(x)=x ALORS σ(x):=m(t)

* $t=f(t_1,\ldots,t_n)$ et $t'=f(t'_1,\ldots,t'_n)$

ALORS E-FILTRE(t_1,t'_1)

...

E-FILTRE(t_n,t'_n)

* SINON échec

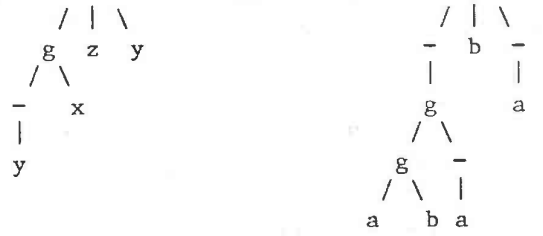
FIN CAS

Preuve: elle résulte des lemmes précédents.

Le coût de l´algorithme est proportionnel à la taille de t. []

REMARQUE : Cet algorithme utilise le fait que la propriété de cohérence sur les termes en E-forme normale peut être assurée par une méthode de descente récursive. On retrouvera le même phénomène pour l´unification.

EXEMPLE : Soient t = et t´ =



Un ensemble complet de E-filtre de t et t´ est { σ = ((y->-a),(x-> g),(z->b))}

comme on peut le vérifier facilement.

5.2- UN ALGORITHME DE E-UNIFICATION DANS AS

5.2.1- GENERALITES

Deux termes t et t´ étant donnés, on cherche à résoudre l´équation (t=t´) dans AS modulo E, c´est à dire trouver les substitutions σ telles que $\sigma(t) =_E \sigma(t')$. Une telle substitution est appelée E-solution de (t = t´).

Comme nous l'avons annoncé dans l'introduction à ce chapitre, nous allons donner un formalisme analogue à celui développé par Martelli et Montanari [M&M,79]. Pour cela, comme on sait associer un système de réécriture canonique à E, la première idée est de transformer une équation donnée en une équation canonique équivalente sur laquelle on va travailler.

DEFINITION 5.1: Deux équations e et e' sont E-équivalentes si et seulement si elles ont même ensemble de E-solutions.

NOTATION: Nous appellerons $\{ -x \mid x \in V \}$ l'ensemble des variables signées et nous le noterons V^-.

LEMME 5.4 : L'équation (t=t') est E-équivalente à l'équation (pfn(t) = pfn(t')).

Preuve: On a $t =_E pfn(t)$ et donc

$\forall \sigma, \sigma(t) =_E \sigma(t') \iff \sigma(pfn(t)) =_E \sigma(pfn(t'))$. []

Nous ne considèrerons donc maintenant que des équations dont les termes sont en forme normale pour E.

Il est naturel de chercher à décomposer une équation donnée en un système d'équations plus faciles à résoudre séparément.

DEFINITIONS 5.2 :On appelle système d'équations S tout multi-ensemble d'équations. On appelle E-solution d'un système d'équations $S=\{(t_i=t_i')_{i=1,..,n}\}$ toute substitution σ E-solution de chacune des équations de S. On dira que les systèmes d'équations S et S' sont E-équivalents si et seulement ils ont même ensemble de E-solutions.

LEMME DE COHERENCE 5.5 : Soit f un symbole d'arité k non nul, $f \neq -$. L'équation $f(t_1,...,t_k)=f'(t_1',...,t_{k'}')$ n'a pas de solution si f est différent de f' et sinon elle est E-équivalente au système $\{(t_i=t_i')_{i=1,..,k}\}$

Preuve: pour toute substitution σ :

$\sigma (f(t_1,\ldots,t_k)) =_E \sigma (f(t_1',\ldots,t_k'))$

$<=> f(\sigma (t_1),\ldots,\sigma (t_k)) =_E f(\sigma (t_1'),\ldots,\sigma(t_k'))$

$<=> pfn(f(\sigma (t_1),\ldots,\sigma (t_k))) = pfn(f(\sigma (t_1'),\ldots,\sigma (t_k')))$

$<=> f(pfn(\sigma (t_1)),\ldots,pfn(\sigma (t_k))) = f(pfn(\sigma (t_1')),\ldots,(pfn(\sigma (t_k')))$

$<=> \forall\ i\ 1 \leq i \leq k => \sigma (t_i) =_E \sigma (t_i')$

<=> σ est solution du système $\{(t_i = t_i')_{i=1,..,k}\}$. []

Par conséquent en itérant ce processus on peut décomposer une équation donnée en un système E-équivalent d'équations telles que

-soit l'une au moins des équations du système est de la forme $(x_i=t_i)$ avec $x_i \in VUV^-$

-soit une équation n'est pas décomposable et alors l'équation $(t=t')$ n'a pas de solutions.

On est donc amené à déterminer l'ensemble des solutions des équations du type $(x_i=t_i)$ avec $x_i \in VUV^-$ et t_i élément de AS.

LEMME 5.6 :Soit $(x=t)$ une équation telle que $x \in V$ et $t \in AS\backslash\{-x\}$. Elle a un ensemble complet minimal de E-unificateurs (on notera $ECMU_E$) Σ

- $\Sigma = \emptyset$ si $x \in V(t)$
- $\Sigma = \{(x-->t)\}$ sinon.

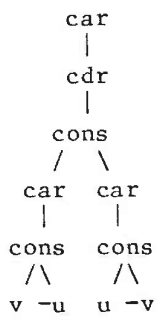
Preuve: elle est évidente. []

Il reste à étudier les équations du type $(x=-x)$ où x est une variable, qui ont clairement pour solutions tous les termes t qui sont E-égaux à leur propre miroir, c'est-à-dire tels que $m(t)=_E t$. C'est pourquoi nous allons maintenant étudier cet ensemble.

DEFINITION 5.3: On note Mir(F) l'ensemble des termes de AS qui sont E-égaux à leur miroir.

$$Mir(F) = \{t \mid t \in AS \text{ et } m(t) =_E t\}$$

EXEMPLE: Si F={cons, car, cdr} les arbres suivants sont elements de Mir(F).



REMARQUES : ** Aucune variable signée ou non, n'est dans Mir(F).

** Si F ne comporte que des symboles d'arité impaire alors Mir(F)=$\emptyset$. En effet, dans ce cas si t $\in$ Mir(F) alors t est fini et non variable, donc il existe u$\in$D(t) tel que

$t_{|u} = f(x_1,\ldots,x_{2p+1})$ avec $\forall i\ 1\leq i\leq 2p+1 \Rightarrow x_i \in VUV^-$ et $t_{|u} =_E m(t_{|u})$
ce qui est impossible car $m(x_p) \neq_E x_p$.

** Si F comporte au moins un symbole d'arité paire et un symbole d'arité impaire alors Mir(F) est infini.
En effet si f_{2q} et f_{2p+1} sont ces symboles
$\{(f_{2p+1})^i(x_1,\ldots,x_{p-1},f_{2q}(x_1,..,x_{q-1},-x_{q-1},..,-x_1),-x_{p-1},\ldots,-x_1) \mid i\in N\} \subseteq$ Mir(F)

Dans la suite nous considérerons F partitionné en l'ensemble des symboles d'arité paire noté FP et en l'ensemble des symboles d'arité impaire noté FI: F=FPUFI.

Afin de pouvoir décrire un ensemble minimal complet de E-unificateurs de l'équation (x=-x) nous allons introduire des sous-ensembles minimaux de Mir(F) notés $Mir_o(F,X)$, où X désigne un sous-ensemble de l'ensemble des variables V. Mir(F,X) va permettre d'obtenir les autres termes de Mir(F) par des instanciations appropriées.

DEFINITION 5.4 : Soit K l'arité maximale des symboles de F, X un sous ensemble de V et V_o le sous ensemble de X suivant:

$$V_o=\{x_i^j \mid j\in N,\ 1\leq i\leq K,\ \text{et } x_i^j\in X\}$$

on pose

$Mir_o^1(F,X)=\{t \mid t\in Mir(F)$ et $|t|=1$ et $t=f(x_1^1,..,x_{p-1}^1,-x_{p-1}^1,..,-x_1^1)$ avec $f\in FP.\}$

....

$Mir_o^n(F,X)=\{t \mid t\in Mir(F)$ et $|t|=n$ et $t=f(x_1^n,..,x_{p-1}^n,t',-x_{p-1}^n,..,-x_1^n)$ avec $f\in FI$ et $t'\in Mir_o^{n-1}(F,X)\}$

$Mir_o(F,X)= \cup_{n\in N} Mir_o^n(F,X)$

LEMME 5.7 : L'équation (x=-x) a un ensemble minimal complet de E-unificateurs

$$\Sigma =\{\sigma \mid \sigma =(x\text{---}>t) \text{ avec } t\in Mir_o(F,X)\}$$

Preuve: ** Si Mir(F) est vide c'est clair.

** sinon vérifions que Σ est bien un $ECMU_E$ de x et -x.

a) $\Sigma \subseteq U_E(x,-x)$ car $Mir_o(F,X) \subseteq Mir(F)$

b) Pour prouver la complétude, il faut montrer que

$\forall\ \alpha\ \in U_E(x,-x),\ \exists\ \sigma \in \Sigma$ tel que $\sigma \leq_E \alpha\ \ [\{x\}]$

Pour cela on va chercher à tronquer le terme $t=\alpha(x)$ de telle sorte le terme obtenu soit plus petit que t. C'est ce que réalise la fonction C de Mir(F) sur $Mir_o(F,X)$ par:

$C(f(t_1,\ldots,t_n))$ = SI $f\in F_{2p+1}$

ALORS SI $|C(t_p)|=k$

ALORS $f(x_1^{k+1},..,x_{p-1}^{k+1},C(t_p),-x_{p-1}^{k+1},..,-x_1^{k+1})$

SINON $(f\in F_{2p})$ $f(x_1^1,..,x_{p-1}^1,-x_{p-1}^1,..,-x_1^1)$ FSI

Soit $t=\alpha(x)$, $t'=C(t)$ et ρ tel que $\rho(t')=t$ (la définition de ρ découle de celle de C); on a alors, en prenant $\sigma=(x\text{---}>t')$

$\rho(\sigma(x))=\alpha(x)$ donc $\sigma \leq_E \alpha\ [\{x\}]$.

c) La minimalité découle de la définition de $Mir_o(F,X)$. []

Par conséquent une équation (t=t') aura en général un $ECMU_E$ infini dont une description finie va être donnée dans ce chapitre.

L'équation (x=-x) ayant une infinité de solutions principales, cela ne permet pas de concevoir un algorithme d'unification "brutal" consistant à travailler par systèmes équivalents en substituant dans toutes les équations du système la solution de l'une d'entre elles.

Nous sommes donc amenés à envisager une approche plus fine dans laquelle, en particulier, la forme des substitutions solutions de l'équation (x=-x) ne sera demandée que pour terminer.

5.2.2- MULTIEQUATIONS

L'idée reprise ici est celle de l'algorithme d'unification de Martelli et Montanari [M&M,79]. L'étude qui suit constitue en quelque sorte une généralisation de leur algorithme à un type bien précis de E-unification. Une telle généralisation semble tout-à-fait possible pour des théories ayant les mêmes caractéristiques que la nôtre, en particulier l'existence d'un système de réécriture canonique associé à E et la conservation des E-solutions d'une équation par les transformations que nous allons décrire. Nous reviendrons sur ce point dans la conclusion de ce chapitre.

DEFINITION 5.5 :Nous appellerons multiéquation e la donnée

-- d'un sous-ensemble fini de V noté V(e)

-- d'un sous-ensemble fini de V^- noté $V^-(e)$

-- d'un multiensemble de termes non variables c'est-à-dire éléments de $AS\setminus(VUV^-)$, noté T(e).

La multiéquation e sera aussi notée $(V(e)=V^-(e)=T(e))$ ou bien encore si

$V(e)=\{x_1,\ldots,x_k\}$, $V^-(e)=\{-x_{k+1},\ldots,-x_p\}$, $T(e)=\{t_1,\ldots,t_q\}$

$$e: (x_1=\ldots=x_k=-x_{k+1}=\ldots=-x_p=t_1=\ldots=t_q)$$

EXEMPLE: (x=z=-x=-y=cons(a,z)=h(z)=cons(a,z))

DEFINITIONS 5.6: ** On appelle système S de multiéquations tout multiensemble de multiéquations. On notera V(S) l'ensemble des variables apparaissant dans S.

** On appelle E-solution d'une multiéquation e toute substitution σ telle que pour tous éléments x, y, t de V(e), $V^-(e)$ et T(e) respectivement on ait $\sigma(x) =_E \sigma(y) =_E \sigma(t)$.

** Des systèmes de multiéquations sont dit E-équivalents s'ils ont même ensemble de E-solutions.

Nous allons maintenant étudier des transformations de systèmes de multiéquations qui simplifient l'expression du système tout en conservant l'ensemble des E-solutions.

5.2.3- DECOMPOSITION D'UN SYSTEME DE MULTIEQUATIONS

L'idée de base, une mutiéquation e étant donnée, est que si les termes de e sont E-unifiables, alors ils possèdent un squelette commun et un système de multiéquations aux différences définis de la facon suivante.

NOTATIONS : Nous noterons $MUL(VUV^-)$ la classe des multiensembles dont les éléments sont les éléments de VUV^-. Lorsqu'il y a possibilité de confusion avec un ensemble, un multiensemble sera noté " Multi{ ... } ".

DEFINITION 5.7: Un multiensemble $ME = \{t_1,\ldots,t_n\}$ de termes de AS étant donné, on définit simultanément

-- une application $SQ_{[ME]}$ d'un domaine d'arbres $D(SQ_{[ME]})$ dans $FUMUL(VUV^-)$ appelée squelette de ME,

-- un système de multiéquations $EqD_{[ME]}$ appelé système de multiéquations aux différences,

de la facon suivante:

a) $D(SQ_{[ME]}) = \bigcap_{1\leq i\leq n} D(t_i)$

b) $\forall m \in D(SQ_{[ME]})\ [\forall i,\ 1\leq i\leq n,\ t_i(m) \in VUV^-]$

=> $SQ_{[ME]}(m)$ est le multiensemble des $t_i(m)$ tels que $t_i(m) \in VUV^-$

et $e=\{t_i(m) \mid t_i(m) \in V\}=\{t_i(m) \mid t_i(m) \in V^-\}=Multi\{\ t_{i|m} \mid t_i(m) \notin VUV^-\ \}$

appartient à $EqD_{[ME]}$.

c) $\forall m \in D(SQ_{[ME]})$ $[\forall i,\ 1 \leq i \leq n,\ t_i(m)=f \in F\setminus\{-\}] \Rightarrow SQ_{[ME]}(m)=f$

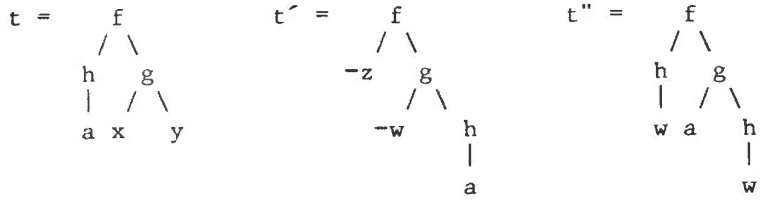
d) s'il existe un élément m de $D(t_i)$ tel que

$(\forall i,\ 1 \leq i \leq n \Rightarrow t_i(m) \in F)$ et $(\exists (i,j),\ 1 \leq (i,j) \leq n$ et $t_i(m) \neq t_j(m))$

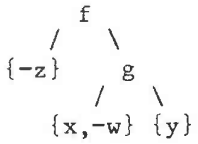
alors SQ et EqD n'existent pas pour ME.

Cette définition technique correspond à une intuition très simple que nous allons illustrer maintenant

EXEMPLE : Si ME est réduit aux trois arbres suivants:



alors $SQ_{[ME]}$ =



et $EqD_{[ME]}$ = $\{(-z = h = h),\ (x = -w = a),\ (y = h = h)\}$

REMARQUES : Le squelette peut très bien ne pas exister au cas où deux symboles de fonctions ne coincident pas. C'est ce qu'exprime la condition b ci-dessus. D'autre part, il est plus simple de considérer un squelette comme étant un terme; c'est ce que nous faisons en considérant en quelque sorte un représentant de SQ de la facon suivante:

A l'application $SQ_{[ME]}$, on associe un terme noté $sq_{[ME]}$ encore appelé squelette, défini par

a) $D(sq_{[ME]})=D(SQ_{[ME]})$

b) $\forall m \in D(sq_{[ME]}),\ SQ_{[ME]}(m) \in F \Rightarrow sq_{[ME]}(m)=SQ_{[ME]}(m)$

c) $\forall\ m \in D(sq_{[ME]})$ $SQ_{[ME]} \in MUL(VUV^{-})$

$\Rightarrow sq_{[ME]}(m)=SI\ \exists\ x \in V \cap SQ_{[ME]}(m)$ ALORS x

SINON un élément quelconque de $SQ_{[ME]}(m)$

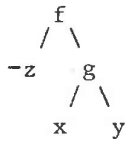
d) enfin si $SQ_{[ME]}$ n'existe pas alors $sq_{[ME]}$ non plus.

Intuitivement, on conserve donc dans les squelettes les variables non signées, quand c'est possible.

Remarquons que la donnée de sq ne détermine pas SQ, mais il est clair que pour tout multiensemble ME d'éléments de AS, la donnée du couple $(SQ_{[ME]},EqD_{[ME]})$ est équivalente à celle de $(sq_{[ME]},EqD_{[ME]})$.
Enfin notons que l'on aurait pu définir également le squelette et le système de multiéquations aux différences de facon inductive.

EXEMPLE : En reprenant l'exemple qui précède, on obtient:

$sq_{[ME]}=$



LEMME 5.8 : La multiéquation e : $(V(e)=\bar{V}(e)=T(e))$ est E-équivalente au système de multiéquations noté Eclat(e) et défini par:

$Eclat(e)=Multi\{(V(e)=\bar{V}(e)=sq_{[T(e)]})\} \cup EqD_{[T(e)]}$ si $sq_{[T(e)]}$ existe, sinon e n'a pas de E-solution.

Preuve: C'est clair par définition de Eclat(e). []

DEFINITION 5.8: On appelle développement ou décomposition d'un système de multiéquations S l'opération consistant à remplacer dans S une multiéquation e par Eclat(e) (s'il existe); on obtient donc ainsi un système S' tel que

$S' = (S\setminus\{e\}) \cup (Eclat(e))$ ou encore

$S' = (S\setminus\{e\}) \cup Multi\{(V(e)=\bar{V}(e)=sq_{[T(e)]})\} \cup EqD_{[T(e)]}$

Le lemme précédent permet de prouver:

COROLLAIRE 5.1 : Si S' est un développement de S alors S et S' sont E-équivalents. Si S n'est pas développable (i.e. pour une multiéquation e de S Eclat(e) n'existe pas) alors S n'a pas de solution.

5.2.4- FUSION D'UN SYSTEME DE MULTIEQUATIONS

Comme Martelli et Montanari, il faut alors regrouper les multiéquations d'un système qui ont des ensembles de variables non disjoints. C'est ce que nous allons décrire maintenant. On notera m(ME) = Multi{ m(t) |t ∈ ME }

DEFINITION 5.9: On appelle fusionnement de deux multiéquations e et e' telles que $((V(e) \cup m(V^-(e)))) \cap (V(e') \cup m(V^-(e'))) \neq \emptyset$ la multiéquation Fus(e,e') définie par:

-- Si $V(e) \cap V(e') \neq \emptyset$ ou $V^-(e) \cap V^-(e') \neq \emptyset$ alors

$V(Fus(e,e')) = V(e) \cup V(e')$

$V^-(Fus(e,e')) = V^-(e) \cup V^-(e')$

$T(Fus(e,e')) = T(e) \cup T(e')$

-- sinon par le fusionnement de e et m(e') qui satisfont alors la condition précédente.

REMARQUE : Une équation e et son miroir m(e) sont clairement E-équivalentes. C'est ce qui permet de remplacer e' par m(e') dans la deuxième partie de la définition ci-dessus.

EXEMPLE : Soient e=(x=y=-x=t), e'=(x=-z=t'), e"=(-y=t").
On a alors Fus(e,e')=(x=y=-x=-z=t=t') et Fus(e,e")=(x=y=-x=t=m(t")), et on ne peut pas fusionner e' et e" puisque leurs ensembles de variables sont disjoints.

Prouvons que le fusionnement de deux équations ne change pas l'ensemble de leurs solutions.

LEMME 5.9 : Soient e et e' deux multiéquations telles que

$$((V(e) \cup m(V^-(e)))) \cap (V(e') \cup m(V^-(e'))) \neq \emptyset,$$

le système d'équations { e,e' } est E-équivalent à la multiéquation Fus(e,e').

Preuve: Compte tenu de la remarque précédente, on ne va étudier que le cas où e et e' sont telles que soit $V(e) \cap V(e') \neq \emptyset$, soit $V^-(e) \cap V^-(e') \neq \emptyset$. Si ce n'est pas le cas, on s'y ramène en étudiant le système { e,m(e') } qui est E-équivalent à { e,e' }.

-- Soit σ une E-solution de e et e' et z un élément de $V(e) \cap V(e')$ ou $V^-(e) \cap V^-(e')$.

$\forall x \in V(e)$, $\forall(-y) \in V^-(e)$, $\forall t \in T(e)$, $\sigma(z) =_E \sigma(-y) =_E \sigma(t)$

et $\forall x' \in V(e')$, $\forall(-y') \in V^-(e')$, $\forall t' \in T(e')$, $\sigma(z) =_E \sigma(-y') =_E \sigma(t')$

donc par transitivité de la E-égalité

$\forall x \in V(e) \cup V(e')$, $\forall(-y) \in V^-(e) \cup V^-(e')$, $\forall t \in T(e) \cup T(e')$

$$\sigma(z) =_E \sigma(x) =_E \sigma(-y) =_E \sigma(t)$$

donc σ est E-solution de Fus(e,e').

-- La réciproque est tout aussi immédiate par définition de Fus(e,e').[]

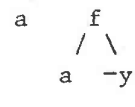
DEFINITION 5.10: On appelle fusion d'un système de multiéquations S et on note Fus(S) l'ensemble des multiéquations obtenu en remplacant dans S tous les couples (e,e') de multiéquations telles que $((V(e) \cup m(V^-(e))) \cap (V(e') \cup m(V^-(e'))) \neq \emptyset$ par leur fusionnement Fus(e,e'). Fus(S) est dit fusionné.

COROLLAIRE 5.2 : Pour tout système S de multiéquations S et Fus(S) sont E-équivalents

5.2.5- ETUDE DES MULTIEQUATIONS TELLES QUE $V(e) \cap m(V^-(e)) \neq \emptyset$

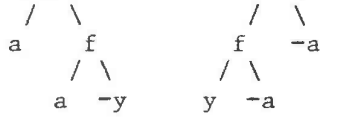
L'objectif du présent paragraphe consiste à étudier une expression particulière des multiéquations sur AS, qui nous permettra d'intégrer dans le formalisme de résolution les solutions des équations (x=-x). Considérons la multiéquation suivante.

EXEMPLE 5.5.1: (x=z=-x= cons) (1)



Elle est E-équivalente, comme nous allons le montrer, à celle qui suit dont l'avantage est de permettre la poursuite du processus de décomposition-fusion:

(x=z= cons = cons) (2)



LEMME 5.10 : Si e est une multiéquation telle que $V(e) \cap m(V^-(e)) \neq \emptyset$ et $T(e) \neq \emptyset$ alors e est E-équivalente à e´ telle que

** $V(e') = V(e)$

** $m(V^-(e')) = m(V^-(e)) \setminus (V(e) \cap m(V^-(e)))$

** $T(e') = T(e) \cup m(T(e))$

On a donc $V(e') \cap V^-(e') = \emptyset$.

Preuve: Soit σ une E-solution de e alors comme il existe x dans $V(e) \cap V^-(e)$, et que $T(e) \neq \emptyset$, on a

$\forall t \in T(e)$ $\sigma(x) =_E \sigma(-x) =_E \sigma(t)$

$\Leftrightarrow$ $\sigma(x) =_E m(\sigma(x)) =_E \sigma(t)$

$\Leftrightarrow$ $\sigma(x) =_E \sigma(t) =_E m(\sigma(t)) =_E \sigma(m(t))$

donc σ est solution de e´.

Réciproquement si σ est solution de e´, alors pour tout x dans $V(e')$, tout -y dans $V^-(e')$ et tout t dans $T(e')$ on a

$\sigma(x) =_E \sigma(-y) =_E \sigma(t) =_E \sigma(m(t))$ par conséquent

si x est élément de $V(e) \cap m(V^-(e))$, alors

$\sigma(-x) =_E m(\sigma(x)) =_E m(\sigma(t)) =_E \sigma(t)$, donc σ est E-solution de e.[]

5.2.6- LA DETECTION DES CYCLES

Comme précédemment, on peut introduire une relation d´ordre notée $<$ entre multiéquations afin de détecter les cycles apparaissant dans les substitutions, pendant le processus de décomposition-fusion plutôt qu´à postériori.

DEFINITION 5.11 : Pour deux multiéquations e et e´

$e < e' \Leftrightarrow \exists x \in V(e) \cup m(V^-(e)) \quad \exists t \in T(e')$ tel que $x \in V(t)$.

L´intérêt d´un tel ordre tient à la propriété suivante:

LEMME 5.11 : Soit S un ensemble de multiéquations fusionné. Si S a une solution alors $\overset{*}{<}$ est un ordre strict partiel.

Preuve: Il faut montrer que $\forall e \in S$, $e \overset{*}{<} e$ est faux. Nous noterons s.t la relation de sous-terme: t s.t t' si et seulement si t est sous-terme strict de t'.

S'il existe e ∈ S tel que $e \overset{*}{<} e$ alors il existe une suite de multi-équations $(e_i)_{i=1,\ldots,n}$ telle que $e=e_1<e_2< \ldots <e_n=e$ (1) avec $e_i \in S$. par conséquent il existe des suites $(x_i)_{i=1,\ldots,n}$ et $(t_i)_{i=1,..,n}$ telles que

- $x_i \in V(e_i) \cup m(V^-(e_i))$
- $t_i \in T(e)$
- $i=1,\ldots,n-1 \Rightarrow x_i \in V(t_{i+1})$

Donc si σ est une solution de S on aura ,pour i=1,...,n-1:

** Si $x_i \in V(e_i)$ alors $\sigma(x_i) =_E \sigma(t_i)$ et $\sigma(x_i)$ s.t $\sigma(t_{i+1})$

donc $\sigma(t_i) =_E \sigma(x_i)$ s.t $\sigma(t_{i+1})$

Donc puisque pour tous termes t et t' de AS on a

$t =_E t' \Rightarrow |t|^- = |t'|^-$ (Où $|t|^-$ désigne le nombre de symboles de fonctions de t distincts de -)

$|\sigma(t_i)|^- < |\sigma(t_{i+1})|^-$

** Si $x_i \in m(V^-(e_i))$ alors $\sigma(-x_i) =_E \sigma(t_i)$ et $\sigma(x_i)$ s.t $\sigma(t_{i+1})$

donc $|\sigma(-x_i)|^- = |\sigma(t_i)|^- = |\sigma(x_i)|^- < |\sigma(t_{i+1})|^-$

D'où encore: $|\sigma(t_i)|^- < |\sigma(t_{i+1})|^-$.

Par conséquent dans tous les cas la relation (1) implique

$$|\sigma(x_1)|^- < |\sigma(t_2)|^- < \ldots < |\sigma(x_{n-1})|^- < |\sigma(t_n)|^- \qquad (2)$$

Mais puisque σ est solution de S: $|\sigma(x_1)|^- = |\sigma(t_n)|^-$

ce qui est en contradiction avec (2).

Par conséquent, $\overset{*}{<}$ ne peut pas avoir de cycle. []

Dans ce qui suit nous aurons également besoin de structurer un ensemble de multiéquations de la facon suivante:

DEFINITION 5.12: Soit S un ensemble de multiéquations fusionné. On lui associe le couple (T,R) défini par

** T est une suite finie de multiéquations $(e_i)_{i=1,..,k}$ telles que

a) $\forall i,j\ 1 \leq i < j \leq k \Rightarrow (V(e_i) \cup m(V^{-}(e_i))) \cap V(T(e_j)) = \emptyset$

(i.e. les variables de $V(e_i) \cup m(V^{-}(e_i))$ peuvent seulement apparaitre dans les multiéquations précédant strictement e_i dans T)

b) $\forall i,\ \#(T(e_i)) < 1$

** $R = S \setminus \{e \mid e \in T\}$

(T,R) est appelé ensemble ordonné de multiéquations.

REMARQUE : T est en fait le début de la substitution E-solution de S, R ce qui reste à résoudre. Par définition, il ne peut y avoir de cycles dans T. Le lemme 5.11, associé à la définition précédente, donne une stratégie permettant d'itérer le processus de décomposition-fusion.

COROLLAIRE 5.3 :Si l'ensemble ordonné de multiéquations S=(T,R) a une E-solution et si R n'est pas vide, alors il existe une multiéquation e de R telle que les variables de e n'apparaissent nulle part ailleurs dans R.

Preuve: $\overset{*}{<}$ étant un ordre partiel sur R (qui est fini), il existe au moins un élément maximal pour $\overset{*}{<}$ dans R. Soit e cet élément. Puisque S est fusionné et par définition de <, les variables de $V(e) \cup m(V^{-}(e))$ n'apparaissent nulle part ailleurs dans R. []

5.2.7- NORMALISATION D'UNE MULTIEQUATION

Pour parvenir à répéter les phases de décomposition-fusion jusqu'à l'apparition de multiéquations e telles que $\#T(e)=1$, on va maintenant fixer une forme particulière des multiéquations, puis des systèmes de multiéquations que nous manipulerons dans la suite.

DEFINITION 5.13: Une multiéquation e est dite normalisée si et seulement si elle vérifie l'une des trois conditions suivantes:

** $\bar{V}(e)=\emptyset$

** $\bar{V}(e)\neq\emptyset$ et *** soit $V(e)\cap\bar{V}(e)=\emptyset$ et $V(e)\neq\emptyset$

*** soit $V(e)\cap\bar{V}(e)\neq\emptyset$ et $T(e)=\emptyset$

Le lemme suivant permettra au lecteur de se convaincre, en le combinant avec le lemme 5.10, que toute multiéquation est E-équivalente à une multiéquation normalisée.

LEMME 5.12 : Toute multiéquation e telle que $V(e)=\emptyset$ est E-équivalente à une multiéquation e' telle que $\bar{V}(e')=\emptyset$.

Preuve: Soit $e=[\bar{V}(e)=T(e))]$ cette multiéquation. On a nécessairement $\bar{V}(e)\neq\emptyset$. Posons $e'=[m(\bar{V}(e))=m(T(e)]$; e et e' sont E-équivalentes et e' satisfait la condition demandée. []

EXEMPLES : La multiéquation (1) de l'exemple 5.5.1 n'est pas normalisée alors que (2) l'est.

$(-x=-y=$ cons$)$ n'est pas normalisée tandis que $(x=y=$ cons $)$ l'est et lui est E-équivalente. Enfin, $(x=y=-x=-z)$ est normalisée.

DEFINITION 5.14: Un système de multiéquations S est dit normalisé si et seulement si:

-- il est fusionné

-- tous ses éléments sont normalisés.

PROPOSITION 5.2: Tout système de multiéquations S peut se mettre sous la forme d'un système normalisé E-équivalent S' noté Nor(S) obtenu en fusionnant S puis en normalisant chaque équation de l'ensemble ainsi obtenu.

Preuve: S et S' sont E-équivalents puisque cette propriété est conservée par les opérations de fusion et de normalisation. []

5.2.8- L'ALGORITHME DE E-UNIFICATION

DESCRIPTION DE L'ALGORITHME.

Soit S un ensemble normalisé et ordonné de multiéquations S=(T,R). Pratiquement on initialisera S à $\{(\emptyset=\emptyset=\{t,t'\})\}$ pour trouver un $ECUM_E$ de t et t', T à $\emptyset$ et R à S.

Tout d'abord on va répéter les phases de développement et de normalisation en mettant dans T les multiéquations qui sont résolues; plus formellement:

DEC-NOR(T,R):

TANT QUE $R\neq\emptyset$ FAIRE

SI il existe $e\in R$ maximale pour $\overset{*}{<}$ (choix de e)

ALORS CAS (1) $T(e)=\emptyset$ ALORS $T\leftarrow T+\{e\}$ (ajouter e en queue de T)

$R\leftarrow R\setminus\{e\}$ (enlever e de R)

(2) sq(T(e)) n'existe pas ALORS ARRET (collision de symboles)

(3) SINON $R\leftarrow R\setminus\{e\}UEqD(e)$ (ajouter à R les équations

$R\leftarrow Nor(R)$ aux différences et normaliser)

$T\leftarrow T+(V(e)=\bar{V}(e)=sq(T(e)))$ (ajouter en queue de T)

FCAS

SINON ARRET (échec par cycle) FSI

FIN TANT QUE

A l'issue de cette procédure, on sait que Σ l'ensemble des E-solutions cherchées est soit vide (détection d'un cycle, collision de symboles de fonctions) soit déterminé par T. Il suffit de prendre les multi-équations de T dans l'ordre et de composer leurs solutions. En effet si $e_1, e_2, \ldots, e_n$ sont les multi-équations de T et si $\sigma_1, \sigma_2, \ldots, \sigma_n$ sont des solutions de ces multi-équations, la composée $\sigma_n\sigma_{n-1}\ldots\sigma_1$ sera solution de T et par conséquent de S.

L'algorithme suivant précise cette dernière étape.

Soit W l'ensemble des variables de $(t = t')$ (ou du système de départ en général). On considère une suite de sous ensembles dénombrables de V, $(X_j)_{j\in N}$, tous disjoints les uns des autres et de W, qui intervient pour des raisons techniques de renommage.

SUBST(T): (on a ici $R=\emptyset$ et $T=(e_i)_{i=1,..,n}$)

(si $T(e_i)\neq\emptyset$ on note t_i son unique élément)

(on notera o la composition des substitutions)

$\Sigma \leftarrow \{Id\}$ $j\leftarrow 0$

POUR i=1 A n FAIRE

CAS (1) $T(e_i)=\emptyset$ et $V(e_i)\cap m(V^-(e_i))=\emptyset$ ALORS (Soit $y\in V(e_i)\neq\emptyset$)

$$\Sigma \leftarrow \bigcup_{\sigma\in\Sigma} [\ \mathop{o}_{x\in V(e_i),x\neq y}(x\to y)\ \mathop{o}_{z\in m(V^-(e_i))}(z\to y)]\circ\sigma$$

(2) $T(e_i)=\emptyset$ et $V(e_i)\cap m(V^-(e_i))\neq\emptyset$ ALORS

$$\Sigma \leftarrow \bigcup_{t\in Mir_o(F,X_j)} \left(\bigcup_{\sigma\in\Sigma} [\ \mathop{o}_{x\in V(e_i)\cup m(V^-(e_i))}(x\to t)\ ]\circ\sigma \right)$$

$j\leftarrow j+1$

(3) SINON $(T(e_i)\neq\emptyset$ et $V(e_i)\cap m(V^-(e_i))=\emptyset)$

$$\Sigma \leftarrow \bigcup_{\sigma\in\Sigma} [\ \mathop{o}_{x\in V(e_i)}(x\to t_i)\ \mathop{o}_{y\in m(V^-(e_i))}(y\to m(t_i))]\circ\sigma$$

FIN CAS

FIN POUR

Avant de voir sur un exemple comment fonctionne cet algorithme nous allons prouver sa correction.

THEOREME 5.1 : Les termes t et t' étant donnés les algorithmes ci-dessus appliqués à $S=\{(\emptyset=\emptyset=\{\{t=t'\}\})\}$ terminent et retournent un $ECUM_E$ Σ de t et t'.

Preuve: terminaison -- SUBST termine de facon évidente.

-- DEC-NOR termine car la taille des membres droits des multiéquations est strictement décroissante.

A chaque étape on conserve la E-équivalence de l'ensemble de

multiéquations par définition des opérations de décomposition et de normalisation. Par conséquent DEC-NOR fournit un résultat qui est soit échec soit un ensemble ordonné de multiéquations $(T,\emptyset)$ E-équivalent à celui de départ. Il reste à montrer que SUBST fournit bien à partir de T un $ECMU_E$ de t et t´

Montrons tout d´abord que tout élément σ de Σ est un E-unificateur de t et t´.

Si on suppose que $T=(e_i)_{i=1,\ldots,n}$ alors tout élément σ de Σ peut s´écrire par définition comme le produit des n substitutions σ_1, σ_2, ..., σ_n où chaque σ_i est solution de la multiéquation e_i.

Il faut encore remarquer que le domaine de chacune de ces substitutions σ_i étant contenu dans $V(e_i)UV^-(e_i)$, (on a l´égalité dans les cas (2) et (3) de SUBST), leurs domaines sont disjoints.

Puisque T est E-équivalent au système de départ, il suffit de montrer que σ est E-solution de T; e_i étant une multiéquation quelconque de T il faut calculer pour tout x_i, $-y_i$ de $V(e_i)$ et $V^-(e_i)$ respectivement:

$\sigma_n \ldots \sigma_1(x_i)$; $\sigma_n \ldots \sigma_1(-y_i)$; $\sigma_n \ldots \sigma_1(t_i)$.

Or, par définition de T, les variables de t_i ne peuvent apparaitre que dans les multiéquations e_j telles que $i<j\leq n$, donc les expressions ci-dessus sont égales au suivantes

$\sigma_n \ldots \sigma_i(x_i)$; $\sigma_n \ldots \sigma_i(-y_i)$; $\sigma_n \ldots \sigma_i(t_i)$

qui sont E-égales par définition de σ_i; ce qui prouve que toute substitution de Σ est un E-unificateur de T donc de t et t´.

Montrons maintenant la complétude. Nous nous référerons aux différentes parties de l´algorithme SUBST en les désignant par "cas (1)" (2) ou (3). Soit α un E-unificateur de t et t´, c´est donc un E-unificateur de T. Nous allons montrer par récurrence sur la longueur n de la suite T que pour tout E-unificateur α de T il existe σ élément de Σ tel que

$$\sigma \leq_E \alpha \ [W] \qquad (*)$$

Pour n=1 il n´y a qu´une seule multiéquation dans T:

Dans les cas (1) et (3) le lemme 5.6 permet de conclure.

Dans le cas (2): si $x \in V(e_1) \cap V^-(e_1)$ alors le lemme 5.7 donne la conclusion; sinon soit z un élément de $V(e_1) \cap V^-(e_1)$; il existe ρ tel que $\rho\,\sigma(z) =_E \alpha(z)$ et par conséquent

$\rho\,\sigma(x) = \rho\,\sigma(z)$ car σ E-unifie e_1

$=_E \alpha(z)$

$=_E \alpha(x)$ car α E-unifie e_1

On a donc également (*) dans ce cas.

Supposons maintenant que la propriété soit vraie lorsque T a n-1 éléments. Nous noterons W_i l'ensemble $V(e_i) \cup m(V^-(e_i)) \cup V(T(e_i))$

Si T a n éléments et si α est un E-unificateur de T, alors α E-unifie e_1 et par conséquent il existe ρ tel que

$$\rho\,\sigma_1 =_E \alpha \qquad [W_1]$$

Nous allons prolonger cette relation de facon à montrer que ρ est un E-unificateur des autres équations de T, c'est-à-dire de $e_2,\ldots,e_n$.

** Si σ_1 est déterminé par le cas (1) alors

$I(\sigma_1) = \{y\} \subseteq W_1$

** Si σ_1 est déterminé par le cas (3) alors

$I(\sigma_1) = V(t_1) \subseteq W_1$

Dans ces deux cas on peut prolonger sans ambiguité ρ de la facon suivante: $\rho = \alpha \quad [V \setminus W_1]$ et puisque $D(\sigma_1) \subseteq W_1$ on a $\rho\,\sigma_1 =_E \alpha$.

Par conséquent, α E-unifiant T et $D(\sigma_1) \cap W_j$ étant vide par définition, ρ est également un E-unificateur de $e_2,\ldots,e_n$. On peut donc appliquer l'hypothèse de récurrence:

$\exists \rho'$ tel que $\rho'\,\sigma_n \ldots \sigma_2 =_E \rho \qquad [W_2 \cup \ldots \cup W_n]$ donc

$\rho'\,\sigma_n \ldots \sigma_2\,\sigma_1 =_E \alpha \qquad [W_1 \cup \ldots \cup W_n]$

** Il nous reste donc à montrer qu'une telle relation est vraie si σ_1 est déterminé par le cas (2).

Dans ce cas, remarquons que par définition de $Mir_o(F,X_1)$, on a toujours:

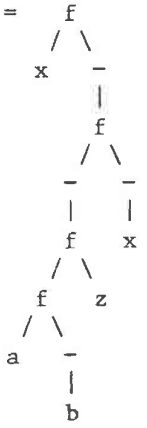
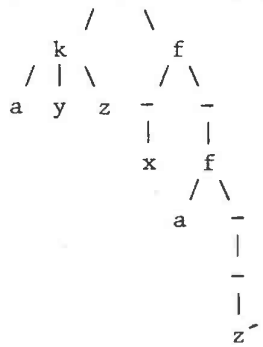
$I(\sigma_1) \cap (W_1 \cup \ldots \cup W_n) = \emptyset$ et par conséquent si on pose

$\rho = \alpha \quad [V \setminus I(\sigma_1) \setminus W_1]$ alors $\rho\sigma =_E \alpha \quad [V \setminus I(\sigma_1)]$

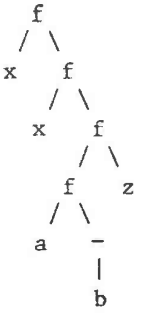
Un raisonnement identique au précédent montre alors que ρ est également un E-unificateur de $e_2,\ldots,e_n$ et par conséquent dans tous les cas on a $\sigma_n \ldots \sigma_1 \leq_E \alpha\ [W_1 U \ldots U W_n]$, ce qui achève la preuve de complétude.

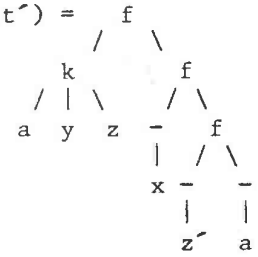
La minimalité découle de la définition des ensembles $Mir_o(F,X_j)$, comme on peut le vérifier facilement. []

Montrons sur un exemple comment fonctionne cet algorithme:

EXEMPLE : Soient t =  et t´ =  à E-unifier.

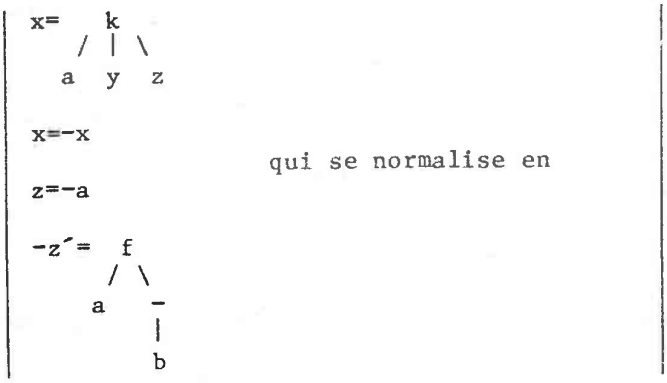
On commence par normaliser t et t´

pfn(t) = 

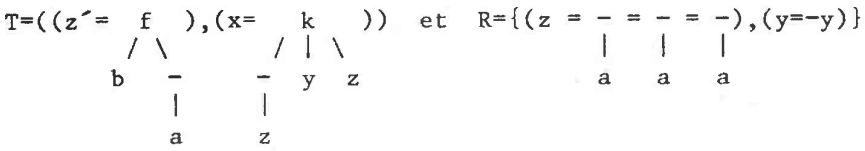
pfn(t´) = 

R est initialisé à { (pfn(t)=pfn(t´)) } et T à $\emptyset$.

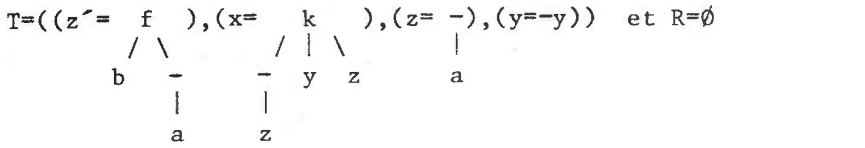
Après une première décomposition, le système obtenu est le suivant:



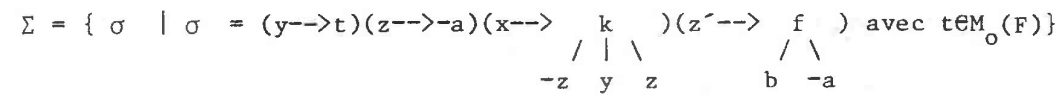
La troisième et la première multiéquations sont maximales pour $\overset{*}{<}$ d´où le nouveau système:



Enfin on obtient:



L´application de SUBST à T fournit alors l´ensemble de E-unificateurs Σ suivant:



CONCLUSION

Une démarche analogue à celle de A.Martelli et U.Montanari nous a permis de résoudre le problème de la E-unification dans la théorie des arbres signés. Il faut remarquer que nous avons utilisé pour cela les mêmes principes que pour résoudre les équations linéaires dans AS, à savoir la transformation d´équations en équations équivalentes plus faciles à résoudre.

Les ensembles complets d'unificateurs obtenus pouvant être infinis, il nous faudra dans la suite en donner un moyen de description finie afin de pouvoir les utiliser. Ceci en particulier dans le but de donner un algorithme complet de A-unification, lorsque A est l'ensemble des axiomes tout entier des algèbres signées.

La méthode que nous avons utilisée peut se généraliser à d'autres théories: il suffit pour cela

* d'une part que l'on puisse définir un squelette et un système de multiéquations aux différences modulo l'ensemble des axiomes E de la théorie, à savoir en reprenant la définition 5.7:

a) $D(SQ_{[ME]})$ est la plus grande partie de $\cap_{1\leq i\leq n} D(t_i)$ contenant ε, close par préfixe, et dont tout élément a par $SQ_{[ME]}$ une image qui n'est pas le symbole de tête d'un axiome de E. Cette première condition permettra de montrer un lemme de cohérence.

b) Les noeuds de ce squelette seront étiquetés par le symbole commun à tous les t_i, (si ce symbole n'est pas commun, le squelette n'existe pas) et ses feuilles d'occurrence m seront constituées du multiensemble des $t_i(m)$. Au système de multiéquations aux différences, il faudra alors ajouter les multiéquations suivantes:

$$e=\{t_i(m) \mid t_i(m)\in V\}=\text{Multi}\{t_{i|m} \mid t_i(m) \text{ est le symbole de tête pour un } i\ (1\leq i\leq n) \text{ d'un axiome de E}\}$$

* d'autre part de donner une méthode appropriée de résolution ou de transformation des multiéquations qu'on ne peut plus décomposer. Par exemple dans notre, cas nous avons donné une transformation (la normalisation) conservant l'ensemble des solutions et permettant de poursuivre le processus de décomposition-fusion. On peut envisager dans d'autres situations d'utiliser une technique de résolution des multiéquations non décomposables telle la surréduction ou la R,E-surréduction.

CHAPITRE SIX

UNE DEUXIEME ETUDE DES ARBRES SIGNES

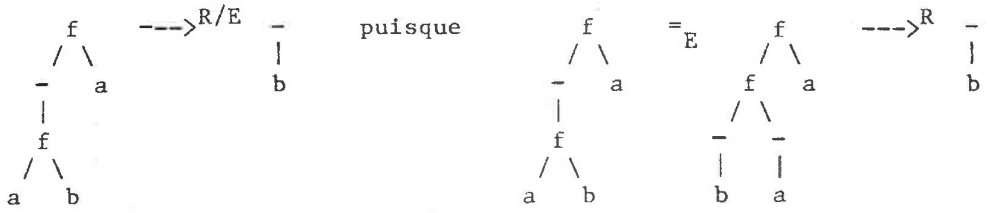
INTRODUCTION

Dans ce chapitre, nous présentons une deuxième étude de la théorie équationnelle engendrée par l'ensemble d'axiomes A = R U E, dans la F-algèbre libre engendrée par V des arbres signés. Notre but est de vérifier les hypothèses nécessaires pour faire de la R,E-surréduction. Pour étudier la R/E-réductibilité, nous utilisons la relation de réécriture $--->^{R,E}$ qui possède la propriété de E-commutation. Nous vérifions enfin que R est un système de réécriture E-canonique.

6.1- ETUDE DE LA REDUCTIBILITE DANS L'ENSEMBLE QUOTIENT

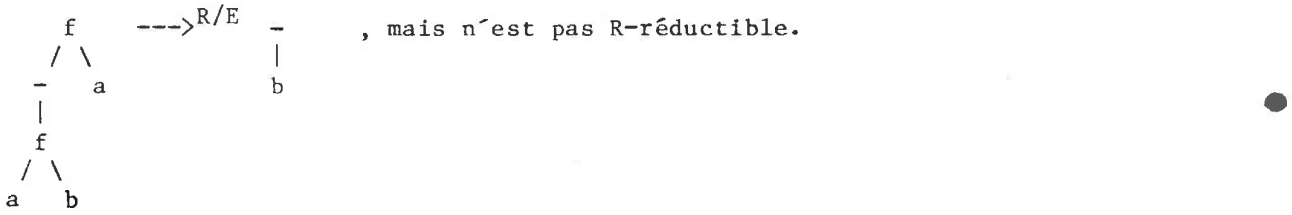
Travailler modulo les axiomes de E suppose la définition d'une relation de réécriture dans l'ensemble quotient notée $--->^{R/E}$ et définie comme la composée de $=_E$ et de $--->^{R}$. Une R/E-réduction est une éventuelle étape de E-égalité suivie d'une R-réduction (voir définition 1.32).

Dans les arbres signés, par exemple



Pour pouvoir en pratique étudier la R/E-réductibilité, il faut disposer d'une relation de réécriture qui soit E-uniforme.

Or, dans AS, la relation $--->^{R}$ obtenue en orientant de gauche à droite les axiomes de (A3), (A3') et (A4) ne vérifie pas la propriété d'E-uniformité, puisque:

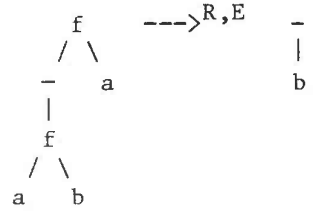


, mais n'est pas R-réductible.

Il faut donc introduire une nouvelle relation de réécriture, déjà utilisée par Peterson et Stickel [P&S,81], notée $--->^{R,E}$ et définie dans le chapitre un (définition 1.46).

Rappelons que la R,E-réductibilité est décidable si l'ensemble de règles R est fini ou décrit finiment et si le E-filtrage est décidable, ce qui est le cas dans AS.

Par exemple



à l'occurrence ε

Nous allons maintenant prouver que la relation $--->^{R,E}$ est E-commutante. Cette propriété est essentielle pour la poursuite de l'étude de la théorie, d'un double point de vue: d'une part, elle va permettre de prouver la E-compatibilité du système de réécriture R et donc la E-confluence, d'autre part, elle intervient aussi dans la construction d'un algorithme complet de A-unification, comme nous l'avons vu dans le chapitre précédent.

6.2- E-COMMUTATION DANS LES ARBRES SIGNES

La proposition 4.3 prouve que la propriété de E-commutation peut se tester sur les paires E-critiques de R/E.

Nous allons maintenant appliquer ce résultat dans AS.

Pour une meilleure lisibilité, nous utiliserons dans le calcul des paires E-critiques la notation arborescente.

Désignons les axiomes et les règles de la facon suivante:

(A) $--x = x$ (A2f) $-f(z_1,\ldots,z_n) = f(-z_n,\ldots,-z_1)$

(A3f) $f(f\#(y)) \longrightarrow y$ (A'3f) $f\#(f(y)) \longrightarrow y$

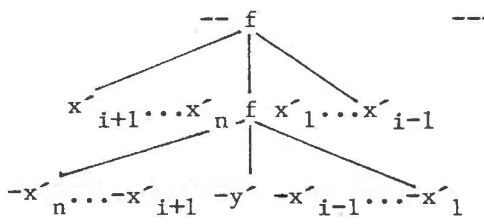
(A4f,i) $f(-x_{i-1},\ldots,-x_1,\ f(x_1,\ldots,x_{i-1},\ y,\ x_{i+1},\ldots,x_n),\ -x_n,\ldots,-x_{i+1}) \longrightarrow y$

en différenciant volontairement les variables des axiomes et des règles.
Le membre gauche de l'axiome (Ai) sera noté g(Ai) et son membre droit d(Ai).

PROPOSITION 6.1: Dans AS, la relation $\longrightarrow^{R,E}$ est localement E-commutante.

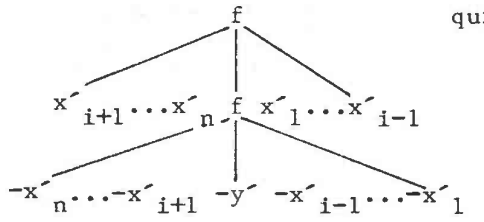
Preuve: la vérification se fait sur un ensemble complet de paires E-critiques de R/E. Une telle paire peut provenir des superpositions suivantes:

* la règle (A4f,i) se superpose dans g(A1) à l'occurrence 1, en donnant:



$\longrightarrow^{R,E}$ $-y'$

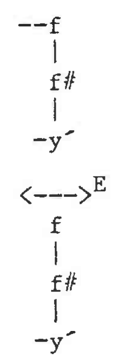
$\longleftarrow^{E}$



qui se R,E-réduit en $-y'$ en utilisant la règle (A4f,n-i).

* la superposition de la règle (A3f) dans g(A1) à l'occurrence 1 conduit

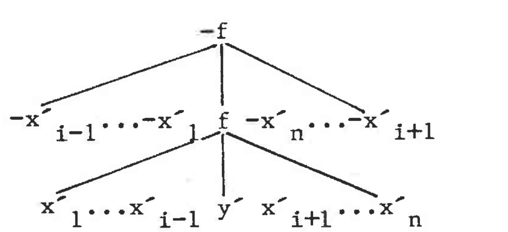
à: $\longrightarrow^{R,E}$ ---y´



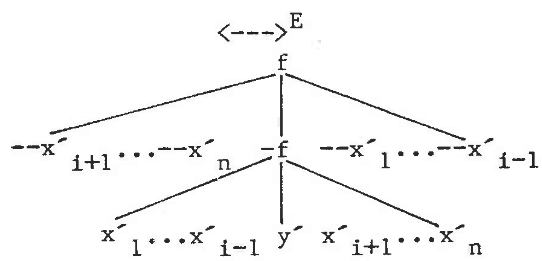
qui se R,E-réduit en $-y' =_E ---y'$

Le cas de la règle (A´3f) se traite de la même manière.

* La règle (A4f,i) se superpose à l´occurrence 1 dans g(A2f):



$\longrightarrow^{R,E}$ -y´

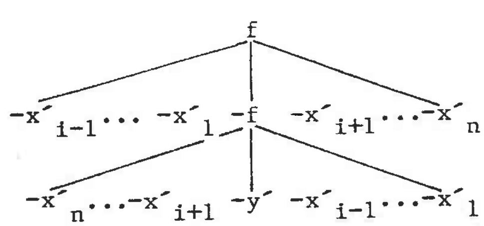


$\longrightarrow^{R,E}$ -y´ en remarquant

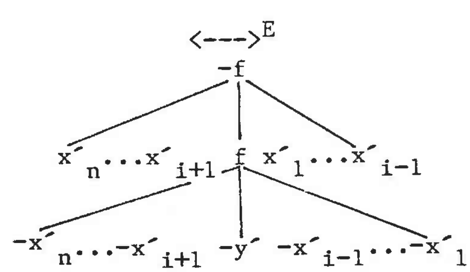
que le deuxième élément de la paire critique obtenue est E-unifiable

avec le membre gauche de la règle (A4f,n-i).

* La règle (A4f,i) se superpose à l´occurrence ε dans d(A2f):



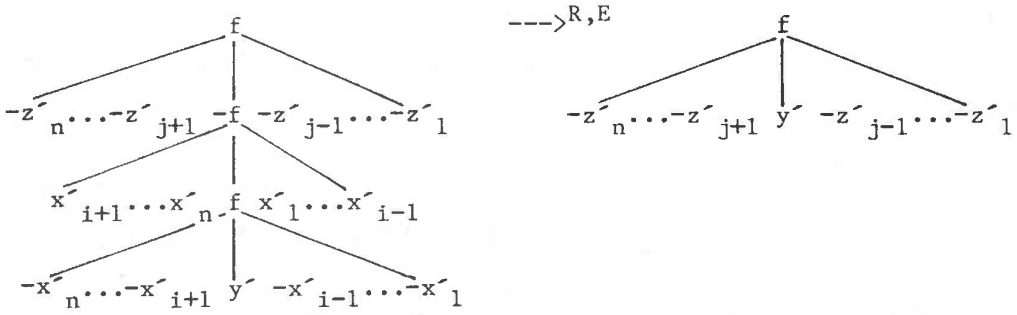
$\longrightarrow^{R,E}$ y´



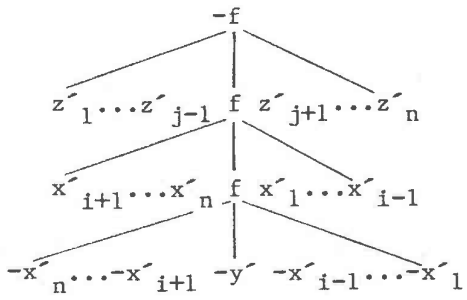
qui est E-unifiable avec

le membre gauche de la règle(A4f,n-i) à l'occurrence 1, et qui se R,E-réduit en $-y' =_E y'$.

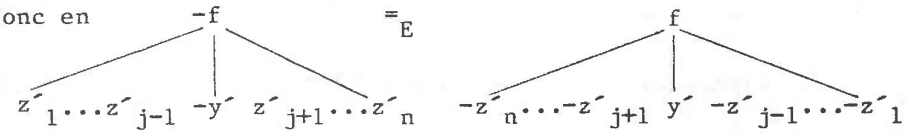
* la règle (A4f,i) se superpose dans d(A2f) à une quelconque occurrence j comprise entre 1 et n en donnant:



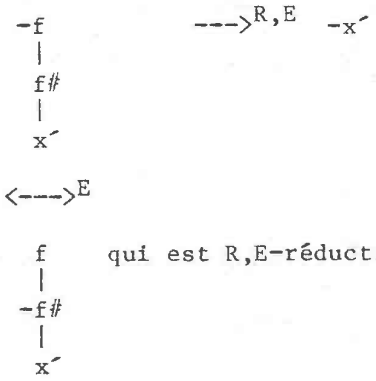
<---->E



dont le sous terme à l'occurrence 1j s' E-unifie avec le membre gauche de la règle (A4f,n-i) et se R,E-réduit donc en



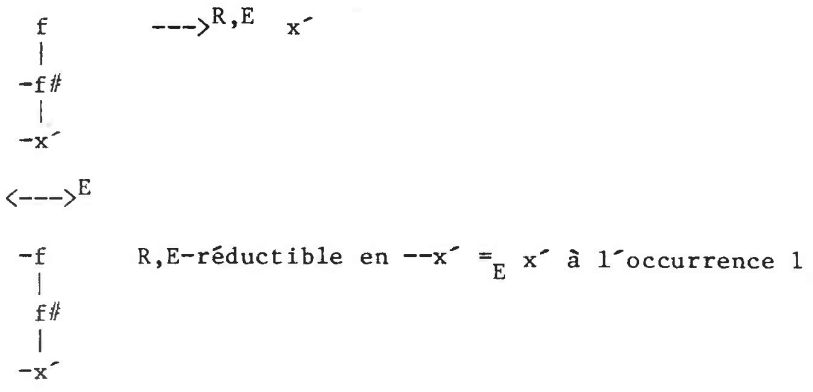
* (A3f) se superpose sur g(A2f) à l'occurrence 1:



qui est R,E-réductible en -x' à l'occurrence ε .

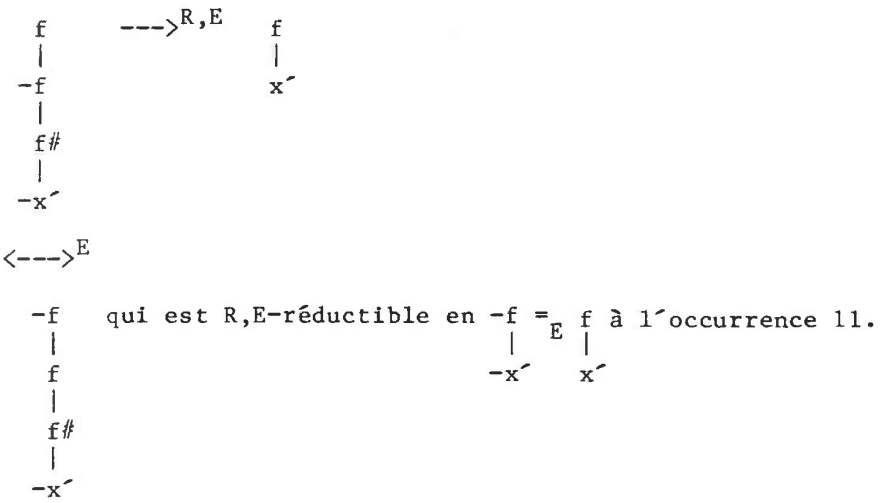
(A'3f) ne se superpose pas à cette occurrence dans g(A2f).

* (A3f) se superpose sur d(A2f) à l'occurrence ε :



(A'3f) ne se superpose pas sur d(A2f) à l'occurrence ε.

* (A3f) se superpose sur d(A2f) à l'occurrence 1:



La superposition de la règle (A'3f) sur d(A2f) à l'occurrence 1 se traite de la même manière.

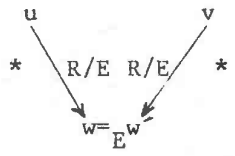
Ayant ainsi vérifié la propriété de E-commutation sur un ensemble complet de paires E-critiques de R/E, nous pouvons en conclure que $--->^{R,E}$ est localement E-commutante donc E-commutante. []

6.3- LA E-CONFLUENCE

La E-confluence de la relation $--->^{R}$ va résulter de la conjonction des trois propriétés suivantes: R est E-clos, E-compatible et E-noethérien.

6.3.1- E-CLOTURE

Rappelons que la preuve de cette propriété consiste à vérifier sur un ensemble complet de paires E-critiques de R que pour toute paire {u,v}



PROPOSITION 6.2: Dans AS, R est E-clos.

Preuve:

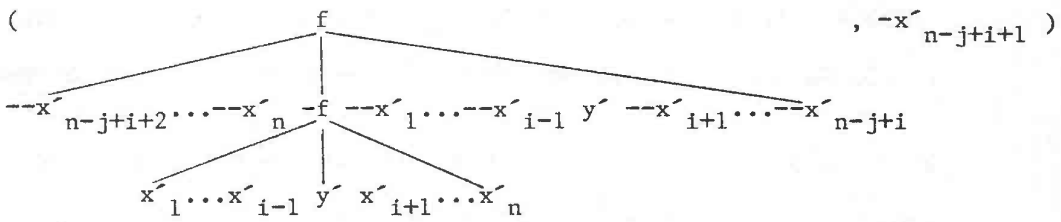
Un E-unificateur σ de deux membres gauches g et g' de règles devant vérifier la condition technique:

$$I(\sigma) \cap (V(g) \cup V(g')) = \emptyset$$

nous utiliserons des variables primées dans les paires critiques.

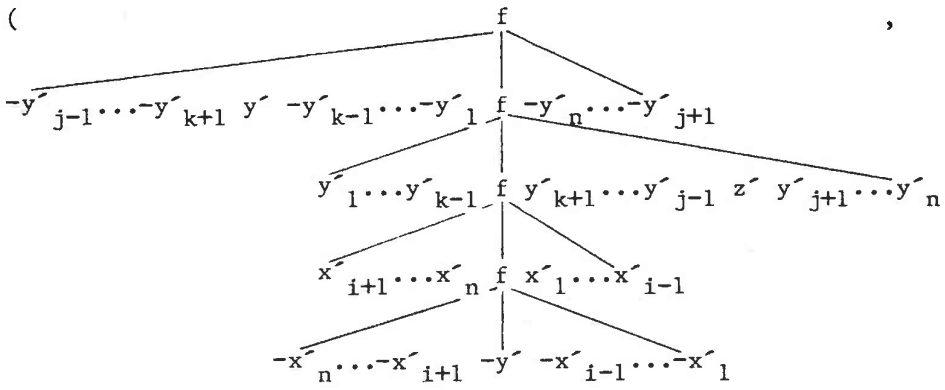
* La règle (A3f) se superpose sur (A'3f) à l'occurrence 1, mais donne une paire E-critique triviale. Il en est de même pour la superposition de (A'3f) dans (A3f) à l'occurrence 1.

* La règle (A4f,i) se superpose dans la règle (A4f,j) à l'occurrence j: la paire E-critique qui en est issue est

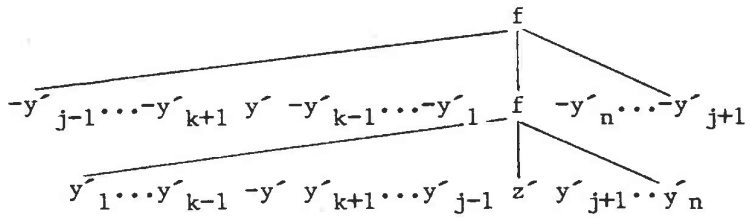


Le premier membre est E-égal au terme obtenu en y remplacant y' par --y' et se réécrit dans R par la règle (A4f,n-j+i+1) en $-x'_{n-j+i+1}$.

* La règle (A4f,i) se superpose dans la règle (A4f,j) à une occurrence k comprise entre 1 et l'arité n de f et différente de j, en donnant la E-paire critique:



En remplacant les x'_j par les termes E-égaux $--x'_j$, on peut appliquer la règle (A4f,n-i) à l'occurrence jk. Le terme obtenu



est E-égal au terme obtenu en remplacant y´ par $--y'$, ce dernier étant alors réductible dans R par la règle (A4f,j) en z´.

* La règle (A4f,i) se superpose dans la règle (A4f,j) à l'occurrence ε : si σ est l'E-unificateur des deux membres gauches de ces règles, il est facile de voir qu'il doit vérifier $\sigma(y) =_E \sigma(y_i)$, $\sigma(z) =_E \sigma(x_j)$ mais aussi $\sigma(y_i) =_E \sigma(x_j)$. La paire E-critique engendrée $(\sigma(y), \sigma(z))$ vérifie donc bien la propriété demandée.

Ayant ainsi envisagé tous les cas possibles de E-superposition, nous pouvons en conclure que R est E-clos. []

Remarquons que les termes à E-unifier ne conduisent jamais à un ensemble complet de E-unificateurs infini. Chacun est réduit à un seul élément et tous les ensembles complets de paires E-critiques sont donc finis.

D'autre part, un algorithme de complétion de Knuth et Bendix généralisé ne générerait pas ici une infinité de règles, puisque toutes les paires E-critiques sont confluentes.

6.3.2- E-COMMUTATION ET E-COMPATIBILITE

LEMME 6.1: Si la relation $--->^{R,E}$ est E-commutante, alors R est E-compatible.

Preuve: soient deux termes t et t_1 tels que $t --->^{R,E} t_1$;
alors il existe un terme t' tel que $t =_E t' --->^R t_1$.
La relation $--->^R$ étant incluse dans $--->^{R,E}$, la propriété de
E-commutation permet d'écrire:

$$\begin{array}{ccc} t' & --->^R & t_1 \\ =_E & & =_E \\ t & --->^{R,E} & t'_1 \end{array}$$

On a alors trivialement le schéma de E-compatibilité:

$$\begin{array}{ccc} t & --->^{R/E} & t_1 \\ \vdots & & \vdots \\ {\scriptstyle R,E}\ \vdots & & *\ \vdots\ {\scriptstyle R/E} \\ \downarrow & & \downarrow \\ t'_1 & =_E & t'_1 \end{array}$$

[]

6.3.3- TERMINAISON FINIE

Il est alors immédiat que la relation $--->^{R/E}$ a la propriété de terminaison finie.

PROPOSITION 6.3: Dans AS, R est E-noethérien.

Preuve: cette propriété résulte clairement du fait qu'une R/E-réduction fait décroitre strictement la taille d'un terme. []

6.3.4- E-CONFLUENCE

Nous utiliserons pour conclure le théorème suivant dû à G.E.Peterson et M.E.Stickel et dont la démonstration peut être trouvée dans [P&S,81]:

PROPOSITION 6.4: Si R est un système de réécriture E-noethérien et E-compatible, alors R est E-confluent si et seulement si il est E-clos.

Jusqu'à présent, cette condition suffisante (mais non nécessaire) de E-confluence a été appliquée à des théories comportant un nombre fini de symboles de fonctions commutatifs et associatifs. L'ensemble des arbres signés en fournit un nouvel exemple d'application.

La conclusion de cette étude se résume de la manière suivante:

PROPOSITION 6.5: Dans AS, le système de réécriture R est E-canonique.

CONCLUSION

Plusieures questions sont suscitées par la double étude de l'ensemble des arbres signés que nous venons de présenter. En effet, à partir de l'ensemble d'axiomes A, l'algorithme de complétion de Knuth et Bendix a généré un ensemble infini de règles. Une première solution étudiée a été l'introduction de "méta-règles" et nous avons montré dans le chapitre deux l'utilité et l'efficacité de ce nouveau concept. Mais l'apparition d'un nombre infini de règles était lié dans AS à l'existence des axiomes de E et la deuxième méthode développée dans ce chapitre a été d'étudier la congruence engendrée par E, puis la confluence des règles issues des axiomes restants dans l'ensemble quotient. On peut alors se demander s'il est possible de relier l'apparition des méta-règles à l'existence d'un tel sous-ensemble des axiomes qui "provoque" l'infinité de règles engendrées. Dans l'affirmative, obtient-on ainsi une nouvelle méthode pour traiter certains systèmes de réécriture infinis, qui sont des cas de divergence de l'algorithme de Knuth et Bendix?
Enfin, E est à l'origine d'un autre problème, celui de l'existence d'un ensemble infini de E-unificateurs pour certaines équations. Là encore, existe-t-il un lien avec ce qui précède? La question reste ouverte.

CHAPITRE SEPT

RESOLUTION D'EQUATIONS QUELCONQUES DANS LES ARBRES SIGNES

INTRODUCTION

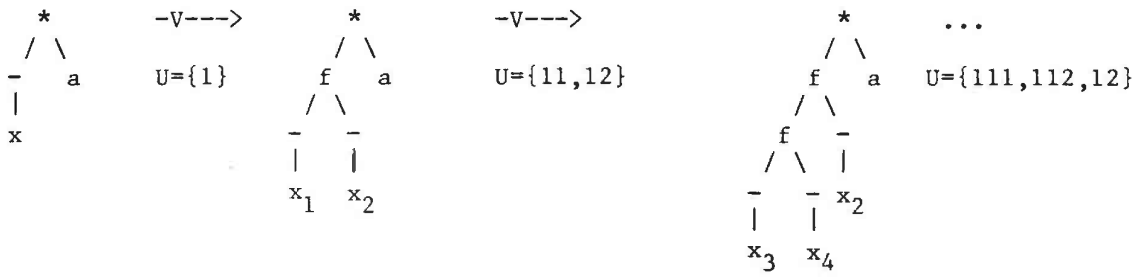
Nous avons présenté dans le chapitre précédent le processus de surréduction permettant de trouver, s'il existe, un ensemble complet de A-unificateurs de deux termes dans une théorie équationnelle A définie par un système de réécriture canonique. Nous avons étendu ces résultats à des théories dans lesquelles l'ensemble des équations initiales est scindé en deux parties: l'une dont toutes les équations (g=d) vérifient V(g)=V(d), l'autre dont les équations sont orientées pour constituer un système de réécriture R. Nous nous proposons maintenant d'appliquer ces résultats théoriques dans l'ensemble des arbres signés muni de l'ensemble d'axiomes A. Disposant de deux facons de décider de la A-égalité, nous nous trouvons devant les deux possibilités suivantes:

-- utiliser le système infini de règles ECR engendré par les méta-règles pour faire de la surréduction,

-- ou partitionner l'ensemble des axiomes en E et R, comme nous l'avons fait dans le chapitre six, et faire de la R,E-surréduction.

Examinons de plus près la première éventualité. En utilisant la surréduction dans ECR, nous nous heurtons au problème de la terminaison d'une surdérivation, même en se limitant aux surréductions basiques. Ce problème vient de l'existence des règles sur le symbole -, comme on peut le voir sur un exemple très simple:

la résolution de l'équation (-x=a) (a étant une constante), en utilisant la surréduction basique, conduit à remarquer que l'ensemble des occurrences basiques, réduit à l'occurrence 1 dans le terme initial, croit indéfiniment dans la surdérivation suivante, où f est un symbole d'arité 2:



Une idée naturelle, lorsqu'on a identifié les axiomes qui "provoquent" ce processus infini, est de faire de la surréduction dans l'ensemble des termes quotienté par la congruence qu'ils engendrent. Chronologiquement, c'est cette idée qui nous a conduit à l'étude présentée au chapitre six.

Rappelons que faire de la R,E-surréduction suppose:

* l'existence d'un algorithme complet de E-unification
* la preuve que R est E-canonique (c'est-à-dire E-confluent et E-noethérien)
* la preuve que la relation $--->^{R,E}$, qui permet d'étudier la réductibilité dans l'ensemble des termes quotienté par $=_E$, est E-commutante.

Toutes ces conditions ont été verifiées aux chapitres cinq et six, dans l'ensemble des arbres signés.

Remarquons enfin qu'en faisant de la R,E-surréduction, notre exemple de surdérivation infinie est trivialement réglé, puisque les deux termes -x et a sont E-unifiables.

La question se pose alors de savoir si toute R,E-surdérivation termine, ce qui sera le cas si on peut se restreindre à des surréductions basiques; en effet, au vu des règles de R, il est clair que l'ensemble des occurrences

basiques va décroitre strictement à chaque étape de R,E-surréduction. Il faudra, pour justifier cette restriction, prouver que la relation $--->^{R,E}$ est strictement E-commutante.

Nous n'aurons pas pour autant prouvé la terminaison de l'algorithme, puisque, dans l'arbre de toutes les surdérivations possibles issues des deux termes à unifier, une infinité de branches peut partir d'un noeud, un ensemble complet de E-unificateurs pouvant être infini. Nous verrons néanmoins qu'il est toujours possible de donner une description finie d'un ensemble complet de A-unificateurs. Ce phénomène est la cause des preuves délicates et techniques de ce chapitre. Il nous semble que des outils appropriés et généraux devraient exister et permettre de simplifier nos preuves. Ce problème n'a pas été abordé.

Enfin nous envisagerons une amélioration possible de cet algorithme en utilisant les résultats obtenus sur les équations linéaires.

7.1- LA R,E-SURREDUCTION DANS LES ARBRES SIGNES

7.1.1- QUELQUES EXEMPLES

Rappelons que E désigne l'ensemble des axiomes suivants:

(A1) = { $--x = x$ }

(A2) = {$-f(x_1,\ldots,x_n) = f(-x_1,\ldots,-x_n)$ pour tout f de F }

et R l'ensemble de règles:

(A3) = { $f(f\#(x)) \longrightarrow x$ pour tout f de $F_1 \cup F'_1$ }

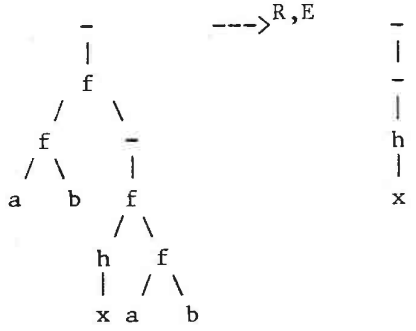
(A'3) = { $f\#(f(x)) \longrightarrow x$ pour tout f de $F_1 \cup F'_1$ }

(A4) = {$f(-x_{i-1},\ldots,-x_1, f(x_1,\ldots,x_i,\ldots,x_n), -x_n,\ldots,-x_{i+1}) \longrightarrow x_i$

notée (4,f,i), pour tout i compris entre 1 et n et tout f d'arité n>1 }.

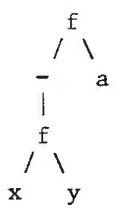
Donnons quelques exemples dans AS de R,E-réduction et de R,E-surréduction.

EXEMPLE :



$\xrightarrow{R,E}$

à l'occurrence 1, par la règle (4,f,2)

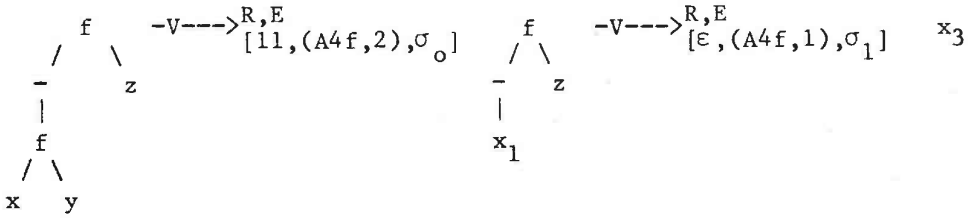


$\xrightarrow{R,E}$

à l'occurrence ε, avec X = {x,y} et σ_X = (x --> a) et la règle (4,f,1)

Donnons également un exemple de R,E-surdérivation:

EXEMPLE : soit X = {x, y, z}



Cette R,E-surdérivation est basique, puisque à chaque étape les ensembles d'occurrences basiques sont les suivants:

$U_0=\{\varepsilon,1,11\}$ $U_1=\{\varepsilon,1\}$ $U_2=\emptyset$

Dans cet exemple, nous avons pris:

σ_0= (x-->x_1)(y--> f) σ_1=(x_1-->x_2)(z--> f)

$-x_1$ x_2 x_2 x_3

7.1.2- LA PROPRIETE DE STRICTE E-COMMUTATION

Afin de justifier le fait de se limiter aux R,E-surréductions basiques, il nous faut prouver que dans AS, la relation $--->^{R,E}$ est strictement E-commutante, c'est-à-dire:

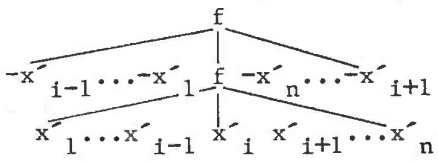
si $t --->^{R,E} t_1$ avec une substitution β R,E-normalisée et si $t =_E t'$, alors il existe un terme t'_1 et une substitution β' R,E-normalisée permettant de R,E-réduire t' tels que

$$t' --->^{R,E} t'_1 =_E t_1$$

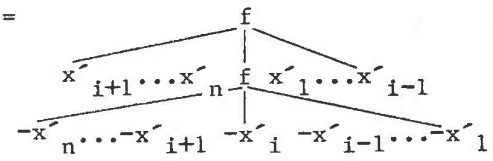
La preuve se fait en considérant $=_E$ comme $<-\underline{n}->^E$ et en faisant une récurrence sur n; le raisonnement ne pose un problème que lorsqu'il y a superposition de la règle avec l'axiome appliqués sur t. Ce sont alors des considérations sur le système de réécriture R qui permettent de conclure. Aussi quelques lemmes techniques préalables, spécifiques à la théorie des arbres signés, sont-ils nécessaires:

LEMME 7.1: Soit $g'--->d'$ une règle de R avec $V(g')=\{x'_i,\ i=1,\ldots,n\}$. Alors il existe une règle $g"--->d"$ de R avec $V(g")=\{x"_i,\ i=1,\ldots,n\}$ et une substitution μ qui est un produit de substitutions $(x"_i-->-x'_j)$ pour i et j compris entre 1 et n, telles que $\mu(g") =_E m(g')$.

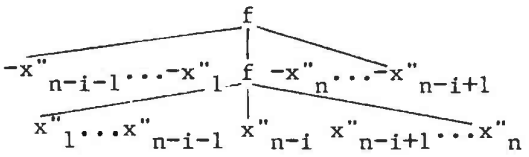
Preuve: Posons $g' =$



$m(g') =$



Il suffit alors de considérer

$g'' =$ 

et la substitution

$\mu = (x''_1 \text{-->} -x'_n)\ldots(x''_{n-i} \text{-->} -x'_i)\ldots(x''_n \text{-->} -x'_1)$. []

LEMME 7.2: Un terme t est R,E-réductible si et seulement si son miroir m(t) est R,E-réductible.

Preuve: supposons t R,E-réductible par la règle $g' \text{--->} d'$ à l'occurrence n avec la substitution σ: $t_{|n} =_E \sigma(g')$. Il existe alors une occurrence n' dans m(t) telle que $m(t)_{|n'} = m(t_{|n}) =_E m(\sigma(g'))$ $=_E \sigma(m(g')) =_E \sigma\mu(g'')$ pour un certain membre gauche g" de règle de R, à cause du lemme précédent. Donc m(t) est R,E-réductible par la règle $g'' \text{--->} d''$ à l'occurrence n'.

La réciproque est immédiate en remarquant que $m(m(t)) =_E t$; en appliquant ce qui précède, m(m(t)) est R,E-réductible et la relation $\text{--->}^{R,E}$ étant E-uniforme, t est donc R,E-réductible. []

LEMME 7.3: Si β est une substitution R,E-normalisée et si μ est un produit de substitutions $(x''_i \text{-->} -x'_j)$ où les x'_i et x''_j sont des variables, alors $\beta\mu$ est R,E-normalisée.

Preuve: il suffit de vérifier que pour toute variable x, $\beta\mu(x)$ est R,E-irréductible. En effet:

soit $\beta\mu(x) = x$ et c'est évident.

soit $\beta\mu(x) = \mu(x) = -x'_j$ et $x'_j \notin D(\beta)$; le résultat est clair.

soit $\beta\mu(x) = \beta(x)$ qui est R,E-irréductible par hypothèse.

soit $\beta\mu(x) = \beta(-x'_j)$ et $x'_j \in D(\beta)$. Dans ce cas, $m(\beta(x'_j))$ est R,E-irréductible par le lemme précédent,

et $\beta\mu(x) =_E m(\beta(x'_j))$ l'est aussi à cause de la propriété de E-uniformité de $--->^{R,E}$. []

Nous sommes alors en mesure de prouver le résultat suivant:

PROPOSITION 7.1: Dans AS, la relation $--->^{R,E}$ est strictement E-commutante.

Preuve: rappelons que l'on souhaite obtenir le diagramme:

$$\begin{array}{ccc} t & \overset{\beta}{--->}_{R,E} & t_1 \\ =_E & & =_E \\ t' & \overset{\beta'}{--->}_{R,E} & t'_1 \end{array}$$

avec β et β' R,E-normalisées

On décompose alors $=_E$ en n étapes $<--->^E$ et on fait une récurrence sur n:

* pour n=0, le résultat est clair.

* pour n=1, on suit de très près la preuve de la proposition 4.3, testant la locale E-commutation sur les paires E-critiques de R/E.
Soit m l'occurrence d'application dans t de l'axiome g=d
n l'occurrence d'application dans t de la règle g'--->d'.

** soit m et n sont disjoints: les substitutions β et β' sont les mêmes.

** il en est de même si $n \leq m$, comme on peut le vérifier aisément dans la démonstration de la proposition 4.3.

** soit $m < n$ et $n=m.m'$

Dans le cas où m' n'est pas une occurrence de O(g), les substitutions utilisées β et β' de la proposition 4.3 sont les mêmes.
Il ne reste donc que le cas où m' est une occurrence de non variable de g différente de ε. Il existe alors une substitution α vérifiant

$$\alpha(g_{|m'}) =_E \beta(g')$$

Considérons alors les différentes possibilités suivantes dans AS:

*** si $g_{|m'} = -x$ avec $x \in V$

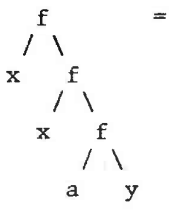
$\alpha(-x) =_E \beta(g')$ implique $\alpha(x) =_E \beta(m(g'))$ et le lemme 7.1 permet de trouver une règle g"--->d" et une substitution μ telles que $\alpha(x) =_E \beta\mu(g")$. Le lemme 7.3 prouve que $\beta' = \beta\mu$ est alors R,E-normalisée.

*** sinon le seul cas possible, au vu des axiomes et des règles dans AS, est celui où $g_{|m'} = f(x_1,\ldots,x_n)$. Remarquons alors que $m(g_{|m'})=d$. De $\alpha(g_{|m'}) =_E \beta(g')$, on déduit $\alpha(m(g_{|m'})) =_E \beta(m(g'))$, puis $\alpha(d) =_E \beta\mu(g")$ pour un certain membre gauche de règle g". Donc $\alpha(d)$ est R,E-réductible à l'occurrence ε par la règle g"--->d" avec la substitution $\beta\mu$ qui est bien R,E-normalisée si β l'est.

* Il est ensuite évident de vérifier que si le résultat est vrai pour n et pour 1, il est vrai pour n+1. []

REMARQUE: Il est clair que l'on aurait pu définir une notion de locale stricte E-commutation et utiliser le fait que cette propriété locale est équivalente à la stricte E-commutation.

La méthode d'unification utilisant la R,E-surréduction basique étant justifiée, toute surdérivation issue de *(t,t'), où (t=t') est l'équation à résoudre, a une longueur finie. Néanmoins , il reste le problème des ensembles infinis de E-unificateurs, illustré par l'exemple suivant:



se R,E-surréduit à l'occurrence ε, avec la règle $f(-x',f(x',y'))\text{--->}y'$ et tout E-unificateur $\sigma = (x\text{-->}t)$ où t est un terme quelconque de $Mir_o(F,X)$.

Nous allons maintenant nous intéresser à ce problème.

7.2- LA META-R,E-SURREDUCTION

Notre but est maintenant de schématiser l'arbre des R,E-surdérivations issues d'une équation à résoudre, en utilisant un processus de méta-E-unification basé sur l'idée suivante: quand deux termes ont une infinité de E-unificateurs, tous ont au moins une composante commune du type (x-->t) où t appartient à $Mir_o(F,X)$. On peut alors schématiser tous ces E-unificateurs en remplacant les composantes de ce type par une composante (x-->z) où z est une "méta-variable", c'est-à-dire une variable qui ne pourra être instanciée qu'en un terme de $Mir_o(F,X)$ à un renommage des variables près.

7.2.1- DEFINITIONS

Soit Z un ensemble dénombrable de variables $\{ z_1, z_2, \ldots, z_n, \ldots \}$ disjoint de V et dont les éléments sont appelés des méta-variables. Pour des raisons techniques de renommage, nous sommes conduits à considérer des ensembles dénombrables de variables X_i tous disjoints et inclus dans V.

DEFINITION 7.1: * Un méta-terme est un terme construit sur F et sur VUZ dont aucun sous-terme n'appartient à un $Mir_o(F,X_i)$.

* Une méta-équation est un couple de méta-termes.

* Une méta-instanciation est une application de Z dans l'ensemble des termes de AS miroirs d'eux-memes.

* Une méta-instanciation φ est principale si et seulement si, pour tout z appartenant à $D(\varphi)$, il existe un ensemble X_i tel que $\varphi(z)$ appartienne à $Mir_o(F,X_i)$, les X_i étant choisis distincts pour chaque z de $D(\varphi)$: donc $V(\varphi(z_i)) \cap V(\varphi(z_j)) = \emptyset$ si $z_i \neq z_j$.

* Une méta-instanciation φ est totale sur un méta-terme t si et seulement si $\varphi(t)$ ne contient plus de méta-variable.

NOTATION: Nous désignerons par ITP(t) (respectivement ITP(u,v)) l'ensemble des méta-instanciations principales totales sur t (respectivement sur les deux méta-termes u et v).

Nous utiliserons dans la suite une propriété technique de ces instanciations principales totales sur t qui justifie l'introduction des ensembles X_i et qui est la suivante:

LEMME 7.4: Pour toute méta-instanciation ψ totale sur le méta-terme t et non principale, il existe un élément φ de ITP(t) et une substitution μ de domaine inclus dans la réunion des X_i, tels que

$$\mu\,\varphi =_E \psi\ [Z \cap V(t)].$$

Preuve: pour tout z appartenant à $Z \cap V(t)$, $\psi(z)$ est un terme de Mir(F) miroir de lui-même. Le b) du lemme 5.7 prouve que pour un tel z il existe des substitutions μ et φ telles que

$$\mu\,\varphi =_E \psi[\{z\}]$$

$\varphi(z)$ appartient à $Mir_o(F,X_i)$ pour un X_i fixé.

En réitérant pour chaque z dans $Z \cap V(t)$ et en prenant chaque fois les X_i disjoints, on définit bien μ et φ vérifiant

$$\mu\,\varphi =_E \psi[V(t)\ Z]. \quad []$$

7.2.2- LA META-E-UNIFICATION

Afin de pouvoir résoudre des méta-équations, il faut disposer d'un algorithme de méta-E-unification, calqué sur l'algorithme de E-unification mais avec la contrainte supplémentaire de se "souvenir" qu'une méta-variable z est solution de l'équation (z=-z). Nous utiliserons dans ce paragraphe les mêmes notations que celles utilisées dans le paragraphe 5.2.8.

7.2.2.1- L'ALGORITHME DE META-E-UNIFICATION

Dans le but de faire ensuite de la R,E-surréduction, nous allons E-unifier des méta-termes u et v en dehors d'un ensemble de variables W contenant V(u)UV(v), c'est-à-dire renommer à chaque fois les variables de $I(\sigma)$, si σ est

le méta-E-unificateur de u et v. Pour désigner le terme t´ obtenu à partir d´un terme t par un renommage de toutes les variables en dehors de W, nous écrirons RENOM(t).

Si u et v sont deux méta-termes, le processus de méta-E-unification est décrit par l´algorithme M-E-UNIF suivant:

M-E-UNIF (u,v)

S <- { (u=v), { $(z_i=-z_i)$ | ∀ z_i ∈ Z∩(V(u)UV(v)) } }

DEC-NOR (∅,S)

M-SUBST(T)

dans lequel DEC-NOR est l´algorithme déjà décrit au paragraphe 5.2.8, $T=(e_i)_{i=1...n}$ est le résultat de DEC-NOR(∅,S) et M-SUBST l´algorithme:

M-SUBST (T)

σ<- Id, W´<- W contenant V(u)UV(v)

POUR i=1 à n FAIRE

CAS (1) $T(e_i)=\emptyset$ et $V(e_i)\cap m(V^-(e_i))=\emptyset$ ALORS

(soit y ∉ W´)

$\sigma \leftarrow \underset{x\in V(e_i)}{o}(x\text{-->}y) \underset{z\in m(V^-(e_i))}{o}(z\text{-->}y) \circ \sigma$

W´<- W´U{y}

(2) $T(e_i)=\emptyset$ et $V(e_i)\cap m(V^-(e_i))\neq\emptyset$ ALORS

(soit $z_i \in Z\backslash W´$)

$\sigma \leftarrow \underset{x\in V(e_i)Um(V^-(e_i))}{o}(x\text{-->}z_i) \circ \sigma$

W´<- W´U{z_i}

(3) SINON ($T(e_i)\neq\emptyset$, $t_i \in T(e_i)$ et

$V(e_i)\cap m(V^-(e_i))=\emptyset$)

$\sigma \leftarrow \underset{x\in V(e_i)}{o}(x\text{-->}RENOM(t_i)) \underset{y\in m(V^-(e_i))}{o}(y\text{-->}m(RENOM(t_i))) \circ \sigma$

W´<- W´UV(RENOM(t_i))

FIN CAS

FIN POUR

DEFINITION 7.2: On appelle méta-E-unificateur des deux méta-termes u et v l'unique substitution σ retournée par M-E-UNIF(u,v).
On notera σ = M-E-UNIF(u,v).

Remarquons que σ ne E-unifie pas en général u et v.
EXEMPLE: L'équation (x=-x) a pour méta-E-unificateur $\sigma = (x \dashrightarrow z_1)$ qui ne vérifie pas $\sigma(x) =_E \sigma(-x)$.

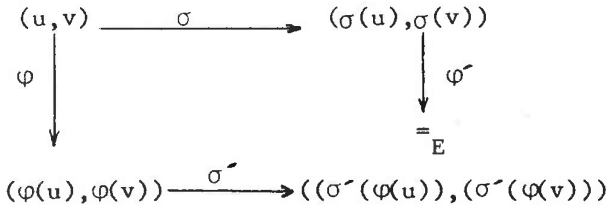
Néanmoins:

LEMME 7.5: L'ensemble $\Sigma = \{ \varphi'\sigma \mid \sigma = \text{M-E-UNIF}(u,v) \text{ et } \varphi' \in \text{ITP}(\sigma(u),\sigma(v)) \}$ est un ensemble minimal complet de E-unificateurs du système d'équations $I = \{ (u=v), \{ (z_i=-z_i) \mid \forall\, z_i \in Z \cap (V(u) \cup V(v)) \} \}$

Preuve: E-UNIF appliqué à I retourne exactement Σ. []

7.2.2.2- E-UNIFICATION ET META-E-UNIFICATION

Nous allons maintenant préciser le lien entre E-unification et méta-E-unification. Les deux lemmes suivants explicitent comment déduire un E-unificateur d'un méta-E-unificateur et réciproquement. Leur preuves, très techniques, peuvent être omises en première lecture et les résultats se résument par le diagramme:



dans lequel $\varphi \in \text{ITP}(u,v)$, σ = M-E-UNIF(u,v), $\varphi' \in \text{ITP}(\sigma(u),\sigma(v))$ et σ' appartient à un ensemble minimal complet de E-unificateur de $\varphi(u)$ et $\varphi(v)$.

LEMME 7.6: Soient u et v deux méta-termes. Si σ = M-E-UNIF(u,v), alors pour toute instanciation principale totale φ' de $\sigma(u)$, $\sigma(v)$

-- $\varphi'\sigma(u) =_E \varphi'\sigma(v)$

-- il existe $\varphi \in$ ITP(u,v) et $\sigma' \in$ ECUM$_E(\varphi(u),\varphi(v))$ tel que

$\sigma'\varphi =_E \varphi'\sigma\ [V(u)UV(v)]$.

Preuve: On veut réaliser le diagramme précédent en partant de

$$(u,v) \xrightarrow{\sigma} (\sigma(u),\sigma(v))$$

* $\forall \varphi' \in$ ITP($\sigma(u)$,$\sigma(v)$), $\varphi'\sigma$ appartient à l'ensemble minimal complet de E-unificateurs de u et v retourné par E-UNIF appliqué au système

$$I \left| \begin{array}{l} u=v \\ (z_i = -z_i) \text{ pour } z_i \in (V(u)UV(v))\cap Z \end{array} \right.$$

donc $\varphi'\sigma$ E-unifie u et v.

* Posons $\psi = \varphi'\sigma_{Z\cap(V(u)UV(v))}$

ψ est une méta-instanciation totale mais non nécessairement principale de u et v.

Par le lemme 7.4, il existe φ appartenant à ITP(u,v) et μ telles que

$$\mu\varphi =_E \psi\ [Z\cap(V(u)UV(v))]$$

Définissons ensuite $\sigma' = \varphi'\sigma_{V\cap(V(u)UV(v))} \cdot \mu$

Vérifions que $\sigma'\varphi =_E \varphi'\sigma\ [V(u)UV(v)]$:

* si $z \in Z\cap(V(u)UV(v))$,

$\sigma'\varphi(z) = \varphi'\sigma_{V\cap(V(u)UV(v))}\mu\varphi(z) =_E \varphi'\sigma_{V\cap(V(u)UV(v))}\psi(z)$

$= \varphi\sigma_{V\cap(V(u)UV(v))}\varphi'\sigma_{Z\cap(V(u)UV(v))}(z) = \varphi'\sigma(z)$.

* si $x \in V\cap(V(u)UV(v))$, $\mu\varphi(x)=x$ et $\sigma'\varphi(x) = \varphi'\sigma(x)$.

* Montrons enfin que σ' appartient à un ensemble minimal complet de E-unificateurs de $\varphi(u)$ et $\varphi(v)$:

Tout d'abord σ' E-unifie bien $\varphi(u)$ et $\varphi(v)$ puisque

$\sigma'\varphi(u) =_E \varphi'\sigma(u) =_E \varphi'\sigma(v) =_E \sigma'\varphi(v)$.

Donc il existe σ_1 appartenant à un ensemble minimal complet de E-unificateurs de $\varphi(u)$ et $\varphi(v)$ tel que

$$\sigma_1 \leq_E \sigma' \ [V(\varphi(u))UV(\varphi(v))]$$

d'où $\sigma_1\varphi \leq_E \sigma'\varphi \ [V(u)UV(v)]$

et $\sigma_1\varphi \leq_E \varphi'\sigma \ [V(u)UV(v)]$.

Comme $\sigma_1\varphi$ est clairement une E-solution du système I et que $\varphi'\sigma$ appartient à un ensemble minimal complet de E-unificateurs de I, on obtient $\sigma_1\varphi =_E \varphi'\sigma \ [V(u)UV(v)]$

$$\sigma_1\varphi =_E \sigma'\varphi \ [V(u)UV(v)]$$

et l'on en déduit, $\varphi(x)$ étant toujours un terme en E-forme normale,

$$\sigma_1 =_E \sigma' \ [V(\varphi(u))UV(\varphi(v))].$$

En conclusion, σ' appartient bien à un ensemble minimal complet de E-unificateurs de $\varphi(u)$ et $\varphi(v)$. []

Un deuxième lemme constitue en quelque sorte la réciproque.

LEMME 7.7: Soient u et v deux méta-termes, φ une méta-instanciation principale totale de u et v telle que $\varphi(u)$ et $\varphi(v)$ soient E-unifiables et σ' un élément d'un ensemble minimal complet de E-unificateurs de $\varphi(u)$ et $\varphi(v)$. Alors

-- u et v sont méta-E-unifiables par le méta-E-unificateur σ.

-- il existe une instanciation principale totale φ' de $\sigma(u)$ et $\sigma(v)$,

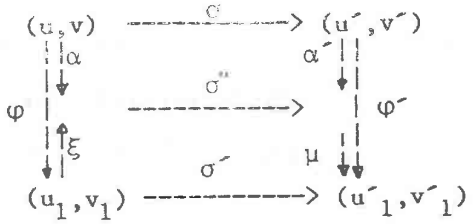
$$\varphi'\sigma =_E \sigma'\varphi \ [V(u)UV(v)].$$

Preuve: Il est clair que u et v sont méta-E-unifiables puisque $\sigma'\varphi$ est une solution de I. D'autre part, il existe α' appartenant à $ITP(\sigma(u),\sigma(v))$ telle que $\alpha'\sigma \leq_E \sigma'\varphi$ et donc une substitution μ vérifiant $\mu\alpha'\sigma =_E \sigma'\varphi \ [V(u)UV(v)]$.

Mais d'après le lemme 7.6, σ et α' étant donnés, il existe α appartenant à ITP(u,v) et α'' appartenant à un ensemble minimal complet de E-unificateurs de $\alpha(u)$ et $\alpha(v)$ telles que

$$\alpha'\sigma =_E \sigma''\alpha \quad [V(u)UV(v)].$$

On peut schématiser la situation par le diagramme suivant:



* α et φ étant deux méta-instanciations principales vérifiant $\alpha'\varphi =_E \mu\sigma''\alpha$ sont forcément comparables sur $Z \cap (V(u) \cup V(v))$, donc on a $\alpha \simeq_E \varphi$ $[Z \cap (V(u) \cup V(v)]$ et il existe alors une bijection ξ telle que $\xi\varphi =_E \alpha$ $[Z \cap (V(u) \cup V(v))]$.

* Comme $\sigma''\alpha \leq_E \sigma'\varphi$ $[V(u) \cup V(v)]$,

$\sigma''\xi\ \varphi \leq_E \sigma'\varphi$ $[V(u) \cup V(v)]$

donc $\sigma''\xi \leq_E \sigma'$ $[V(\varphi(u)) \cup V(\varphi(v))]$.

Mais $\sigma''\xi$ E-unifie $\varphi(u)$ et $\varphi(v)$ et σ' appartient à un ensemble minimal complet de E-unificateurs de ces deux termes, donc

$\sigma''\xi \simeq_E \sigma'$ $[V(\varphi(u)) \cup V(\varphi(v))]$

et il existe une bijection β vérifiant

$\sigma''\xi =_E \beta\sigma'$ $[V(\varphi(u)) \cup V(\varphi(v))]$.

* De $\sigma'\varphi =_E \mu\sigma''\alpha$ $[V(u) \cup V(v)]$ et des résultats précédents, on déduit $\beta^{-1}\sigma''\xi\varphi =_E \mu\sigma''\xi\varphi$ $[V(u) \cup V(v)]$, puis $\mu =_E \beta^{-1}$.

μ étant donc une bijection, $\mu\alpha'$ est une instanciation principale totale de $\sigma(u)$ et $\sigma(v)$; en posant $\varphi' = \mu\alpha'$, on obtient bien

$$\varphi'\sigma =_E \sigma'\varphi \quad [V(u) \cup V(v)]. \quad []$$

7.2.3- LA META-R,E-SURREDUCTION BASIQUE DANS AS

Dans le processus de R,E-surréduction basique, aucune R,E-surréduction n'est faite sur des termes de $Mir_0(F,X_i)$, puisque ceux-ci sont forcément apportés par une substitution. Cette remarque permet d'introduire la méta-R,E-surréduction que nous allons maintenant définir, dans les arbres signés.

7.2.3.1- DEFINITION DE LA META-R,E-SURREDUCTION

DEFINITION 7.3: * Un méta-terme t est méta-R,E-surréductible en t´ à l´occurrence m, avec la règle k et le méta-E-unificateur σ si et seulement si

-- $t_{|m}$ est méta-E-unifiable avec g_k

-- σ = META-E-UNIF($t_{|m}, g_k$)

-- $t' = \sigma(t[m<-d_k])$

et l´on note $t \text{ -V--->>}^{R,E}_{[m,k,\sigma]} t'$.

* Une méta-R,E-surdérivation est une suite de méta-R,E-surréductions issues d´un méta-terme.

* A une méta-R,E-surdérivation issue de *(t,t´)

$$*(t,t') \text{ -V--->>}^{R,E}_{[m_0,k_0,\sigma_0]} *(t_1,t'_1) \quad \ldots \quad \text{-V--->>}^{R,E}_{[m_{n-1},k_{n-1},\sigma_{n-1}]} *(t_n,t'_n)$$

telle que t_n et t'_n soient méta-E-unifiables par μ, on associe la substitution $\mu\sigma_{n-1}\ldots\sigma_1\sigma_0$ appelée méta-solution de la méta-équation(t=t´).

NOTATION: Quand l´occurrence ou la règle utilisées dans une R,E-surréduction ou une méta-R,E-surréduction seront sans importance dans les raisonnements, nous noterons simplement $\text{-V--->}_{[\sigma]}$ ou $\text{-V--->>}_{[\sigma]}$.

Nous nous proposons maintenant de prouver que l´ensemble des méta-solutions permet de trouver un ensemble complet de A-solutions de l´équation (t=t´) si t et t´ sont des termes de AS.

Un premier lemme important est nécessaire pour montrer comment, à partir d´une méta-solution, on retrouve des A-solutions; il permet de construire une étape de R,E-surréduction à partir d´une étape de méta-R,E-surréduction, conformément au diagramme suivant:

$$\begin{array}{ccc} t & \text{-V--->>}^{R,E}_{[m,k,\sigma]} & t' \\ \varphi \downarrow & & \downarrow \varphi' \\ & & t'' \\ & & =_E \\ t_1 & \text{-V--->}^{R,E}_{[m,k,\sigma']} & t'_1 \end{array}$$

où $\varphi \in ITP(t)$ et $\varphi' \in ITP(t')$.

LEMME 7.8: Si t est un méta-terme qui se méta-surréduit en t' à l'occurrence m avec la règle k et le méta-E-unificateur σ, et si φ' est une méta-instanciation principale totale de t', alors il existe une méta-instanciation φ principale totale de t telle que $\varphi(t)$ soit R,E-surréductible à l'occurrence m, avec la règle k et la substitution σ' en un terme E-égal à $\varphi'(t')$.

Preuve: on cherche à réaliser le diagramme précédent à partir de

$$t \ -V--->>^{R,E}_{[m,k,\sigma]} \ t'$$
$$\Big\downarrow \varphi'$$
$$\varphi'(t')$$

Puisque $t_{|m}$ et g_k sont méta-E-unifiables par σ, d'après le lemme 7.6, il existe φ appartenant à $ITP(t_{|m}, g_k)$ et σ' appartenant à un ensemble minimal complet de E-unificateurs de $\varphi(t_{|m})$ et de $\varphi(g_k)=g_k$ telles que

$$\sigma'\varphi =_E \varphi'\sigma \quad [V(t_{|m}) \cup V(g_k)]$$

Il est toujours possible de prolonger φ en une méta-instanciation principale totale sur t, encore désignée par φ.

Alors $\varphi(t) \ -V--->^{R,E}_{[m,k,\sigma']} \ \sigma'(\varphi(t)[m<-d_k])$

$$= \sigma'\varphi(t)[m<-\sigma'\varphi(d_k)]$$
$$=_E \varphi'\sigma(t)[m<-\varphi'\sigma(d_k)] = \varphi'(t'). \quad []$$

7.2.3.2- LA CORRECTION DE LA META-R,E-SURREDUCTION

La proposition suivante prouve la correction du processus de méta-R,E-surréduction.

PROPOSITION 7.2: Soient t et t' deux termes de AS et σ une méta-solution de l'équation (t=t'). Alors pour toute méta-instanciation φ' principale totale de $\sigma(t)$ et $\sigma(t')$, $\varphi'\sigma$ est une A-solution de (t=t').

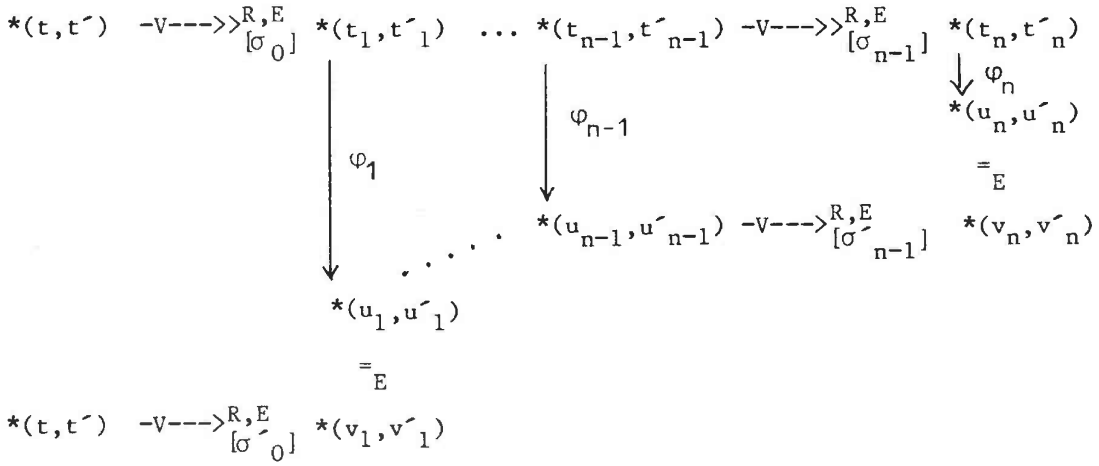
Preuve: la méta-solution σ provient d'une méta-R,E-surdérivation:

$*(t,t') \text{ -V-->> } *(t_1,t'_1) \ \ldots \ \text{-V-->> } *(t_n,t'_n)$.

Les méta-termes t_n et t'_n étant méta-E-unifiables par μ, d'après le lemme 7.6, pour toute φ' appartenant à $ITP(\mu(t_n),\mu(t'_n))$, $\varphi'\mu(t_n) =_E \varphi'\mu(t'_n)$ et il existe φ_n appartenant à $ITP(t_n,t'_n)$ et μ' appartenant à un ensemble minimal complet de E-unificateurs de $\varphi_n(t_n)$ et $\varphi_n(t'_n)$ tels que

$$\mu'\varphi_n =_E \varphi'\mu \quad [V(t_n) \cup V(t'_n)].$$

Le lemme 7.8 permet alors de construire par récurrence le diagrammme:



Montrons alors que, pour toute instanciation principale totale φ' de $\sigma(t)$ et $\sigma(t')$, $\varphi'\sigma$ est une A-solution de l'équation $(t=t')$.

En effet, il est facile de vérifier, en utilisant le fait que, pour tout i compris entre 1 et n, $\varphi_i\sigma_{i-1} =_E \sigma'_{i-1}\varphi_{i-1}$ et $\sigma_0 = Id$, $\varphi'\mu\sigma_{n-1}\cdots\sigma_0(t) =_E \mu'\sigma'_{n-1}\cdots\sigma'_0(t)$. Puis:

$\mu'\sigma'_{n-1}\cdots\sigma'_0(t) =_A \mu'\sigma'_{n-1}\cdots\sigma'_1(v_1)$

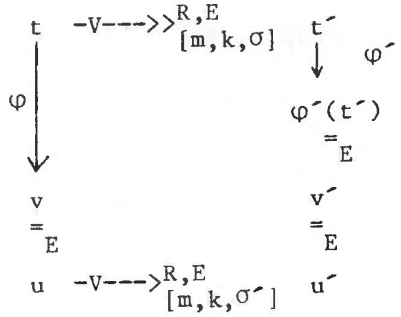
$=_E \mu'\sigma'_{n-1}\cdots\sigma'_1(u_1) = \ldots = \mu'\sigma'_{n-1}(u_{n-1})$

$=_A \mu'(v_n) =_E \mu'(u_n) =_E \mu'(u'_n)$

$=_A \mu'\sigma'_{n-1}\cdots\sigma'_0(t')$ par un raisonnement analogue sur t'. []

7.2.3.3- LA COMPLETUDE DE LA META-R,E-SURREDUCTION

Afin de prouver que toute A-solution d'une équation $(t=t')$ se déduit d'une méta-solution de $(t=t')$, nous démontrons un nouveau lemme important, qui permetde faire correspondre une étape de méta-R,E-surréduction à une étape de R,E-surréduction basique conformément au diagramme suivant:



dans lequel φ et φ' sont des instanciations principales totales.
La preuve de ce lemme peut être omise en première lecture.

LEMME 7.9: Soient t un méta-terme, u un terme de AS, φ une méta-instanciation principale totale sur t telle que $\varphi(t) =_E u$ et B un sous-ensemble d'occurrences clos par préfixe de $dom(t) \cap dom(u)$ vérifiant les deux conditions suivantes:

1) $\forall\, b \in B,\ t(b) = u(b)$.

2) aucune occurrence de dom(t) au dessous d'un symbole - n'appartient à B, c'est-à-dire: si $b \in B$ et $t(b) = -$, alors $b.1 \notin B$.

Si u est R,E-surréductible à une occurrence m appartenant à B en un terme u', par la règle k et avec la substitution R,E-surréductrice σ',
alors t est méta-R,E-surréductible en un méta-terme t' et il existe une instanciation φ' principale totale de t' telle que $\varphi'(t') =_E u'$. De plus,
$B' = B \setminus \{ p \mid p \in B \text{ et } p \geq m \}$ vérifie par rapport à u' et t' les conditions 1 et 2.

Preuve: u étant R,E-surréductible en u', $\sigma'(u_{|m}) =_E \sigma'(g_k)$.

Les propriétés 1 et 2 impliquent que, pour toute occurrence b de B, $u_{|b} =_E \varphi(t)_{|b}$. Donc $\sigma'(\varphi(t_{|m})) =_E \sigma'(g_k)$.

Ce qui prouve que $\varphi(t_{|m})$ est E-unifiable avec g_k par la substitution σ'. Le lemme 7.7 permet alors de prouver que $t_{|m}$ est méta-E-unifiable avec g_k par un méta-E-unificateur σ et qu'il existe φ' appartenant à $ITP(\sigma(t_{|m}),\sigma(g_k))$ telle que

$$\varphi'\sigma =_E \sigma' \varphi[V(t_{|m}) \cup V(g_k)].$$

On en déduit sans peine que

$$t \ -V-->>^{R,E}_{[m,k,\sigma]} \ t' = \sigma(t[m<-d_k])$$

et

$$\varphi(t) \ -V-->^{R,E}_{[m,k,\sigma']} \ \sigma'(\varphi(t)[m<-d_k]).$$

La propriété 1 permet encore d'écrire que

$$\varphi(t)[m<-d_k] =_E u[m<-d_k] \quad \text{et donc que}$$

$$\sigma'(\varphi(t)[m<-d_k]) =_E u'.$$

Enfin φ' peut être étendue en une instanciation principale totale sur $\sigma(t)$ encore notée φ' et l'on obtient:

$$\varphi'(t') = \varphi'(\sigma(t[m<-d_k])) =_E \sigma'(\varphi(t[m<-d_k])) =_E u'.$$

Cette construction se visualise de la facon suivante:

$$\begin{array}{ccc}
t & -V-->>^{R,E}_{[m,k,\sigma]} & t' = \sigma(t[m<-d_k]) \\
\Big\downarrow \varphi & & \Big\downarrow \varphi' \\
 & & \varphi'(t') = \varphi'\sigma(t[m<-d_k]) \\
 & & =_E \\
\varphi(t) & -V-->^{R,E}_{[m,k,\sigma']} & \sigma'\varphi(t[m<-d_k]) \\
=_E & & =_E \\
u & -V-->^{R,E}_{[m,k,\sigma']} & u'
\end{array}$$

Pour terminer, il est clair, par construction de B', que les conditions 1 et 2 sont vérifiées. []

Nous sommes alors en mesure de prouver la complétude du processus de méta-R,E-surréduction.

PROPOSITION 7.3: Pour toute A-solution α de l'équation $(t=t')$, où t et t' sont deux termes de AS, il existe une méta-solution σ et une méta-instanciation φ' principale totale de $\sigma(t)$ et $\sigma(t')$ telles que

$$\varphi'\sigma \leq_A \alpha\ [V(t)UV(t')].$$

Preuve: nous savons que pour toute A-solution α de $(t=t')$, il existe une R,E-surdérivation basique issue de t qui fournit une A-solution σ' telle que $\sigma' \leq_A \alpha\ [V(t)UV(t')]$:

$$*(t,t') \text{ -V---}>^{R,E} *(u_1,u'_1) \ \ldots\ \text{ -V---}>^{R,E} *(u_n,u'_n)$$

A partir de cette R,E-surdérivation basique, nous allons construire une méta-R,E-surdérivation associée

$$*(t,t') \text{ -V---}>>^{R,E} *(t_1,t'_1) \ \ldots\ \text{ -V---}>>^{R,E} *(t_n,t'_n).$$

en utilisant le lemme précédent et en prenant pour B, à chaque étape, l'ensemble des occurrences basiques de la R,E-surdérivation précédente.

En posant $B_0 = O(*(t,t'))$, on obtient, par une application immédiate du lemme 7.9, les méta-termes t_1 et t'_1, une instanciation φ_1 principale totale de $*(t_1,t'_1)$ vérifiant

$\varphi_1(*(t_1,t'_1)) =_E *(u_1,u'_1)$

et un ensemble $B_1 = B_0 \setminus \{ p \in B_0 \text{ et } p \geq m \}$ qui est l'ensemble des occurrences basiques de $*(u_1,u'_1)$.

En itérant cette construction, on construit une méta-R,E-surdérivation conformément au schema:

$$\begin{array}{ccccccc}
*(t,t') & \text{-V---}>>_{[\sigma_0]} & *(t_1,t') & \ldots & \text{-V---}>>_{[\sigma_{n-1}]} & *(t_n,t'_n) \\
 & & \downarrow \varphi_1 & & & \downarrow \varphi_n \\
 & & =_E & & & =_E \\
*(t,t') & \text{-V---}>_{[\sigma'_0]} & *(u_1,u'_1) & \ldots & \text{-V---}>_{[\sigma'_{n-1}]} & *(u_n,u'_n)
\end{array}$$

En appliquant le lemme 7.7, u_n et u'_n étant E-unifiables par une substitution μ', φ_n étant une instanciation principale totale telle que $\varphi_n(*(t_n,t'_n)) =_E *(u_n,u'_n)$, on en déduit que t_n et t'_n sont méta-E-unifiables par μ et qu'il existe une instanciation φ' principale totale de $\mu(t_n)$ et $\mu(t'_n)$ vérifiant $\mu'\varphi_n =_E \varphi'\mu\ [V(t_n) \cup V(t'_n)]$.

Comme, pour tout i compris entre 1 et n,

$\varphi_i\sigma_{i-1} =_E \sigma'_{i-1}\varphi_{i-1}\ [V(t_i) \cup V(t'_i)]$, et que $\varphi_0 = Id$,

on obtient, sur $V(t) \cup V(t')$:

$\varphi'\mu\sigma_{n-1}\cdots\sigma_0 =_E \mu'\varphi_n\sigma_{n-1}\cdots\sigma_0$

$=_E \mu'\sigma'_{n-1}\varphi_{n-1}\sigma_{n-2}\cdots\sigma_0 \cdots =_E \mu'\sigma'_{n-1}\cdots\sigma'_0$.

Puisque $\sigma' = \mu'\sigma'_{n-1}\cdots\sigma'_0 \leq_A \alpha\ [V(t) \cup V(t')]$, on a bien $\varphi'\sigma \leq_A \alpha$. []

7.2.4- DESCRIPTION FINIE D'UN ENSEMBLE COMPLET DE A-SOLUTIONS D'UNE EQUATION

L'arbre de toutes les méta-R,E-surdérivations issues de l'équation $(t=t')$ à résoudre est fini et nous savons construire un ensemble complet de A-solutions à partir des méta-solutions obtenues. La méta-R,E-surréduction dans les arbres signés fournit donc un processus fini, permettant d'engendrer éventuellement un ensemble infini de solutions.

Les résultats précédents se résument ainsi:

THEOREME 7.1: Soient t et t' deux termes de AS et Σ l'ensemble des substitutions σ telles que

* il existe une méta-R,E-surdérivation

$*(t,t') \text{ -V---}>>_{[\sigma_0]} *(t_1,t'_1) \ \cdots\ \text{ -V---}>>_{[\sigma_{n-1}]} *(t_n,t'_n)$

avec t_n et t'_n méta-E-unifiables par μ

* $\sigma = \mu\sigma_{n-1}\cdots\sigma_0$

Alors Σ est un ensemble fini et

$\Sigma' = \{\ \sigma' \mid \sigma' = \varphi'\sigma \text{ avec } \varphi' \in ITP(\sigma(t),\sigma(t'))\ \}$

est un ensemble complet de A-unificateurs de t et t'.

Preuve: elle découle des propositions 7.2 et 7.3. []

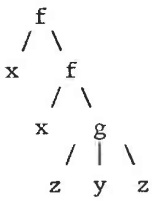
Nous obtenons ainsi un algorithme de A-unification consistant à engendrer l'ensemble de toutes les méta-solutions de l'équation donnée et qui cette fois-ci termine.

REMARQUES: * L'ensemble des solutions de l'équation (t=t') est fini si et seulement si toutes les méta-R,E-surdérivations issues de *(t,t') sont des R,E-surdérivations.

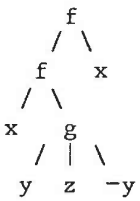
* L'ensemble de A-unificateurs obtenu n'est pas en général minimal et nous verrons dans le paragraphe suivant une amélioration possible utilisant les résultats obtenus sur les équations linéaires.

Auparavant, nous allons développer un exemple de méta-R,E-surdérivation en montrant une R,E-surdérivation associée.

EXEMPLE: f et g désignent des symboles de fonctions, x, y et z des variables, z_1, z_2, z_3 sont des méta-variables.



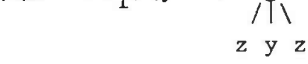
=



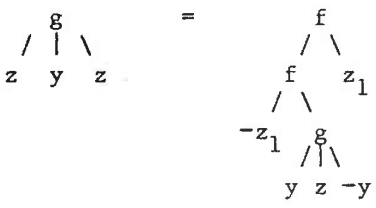
se méta-R,E-surréduit à l'occurrence 1

avec la règle (f(-x',f(x',y'))--->y'

et le méta-E-unificateur

$(x \to z_1)(x' \to z_1)(y' \to$  $)$

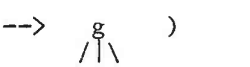
en

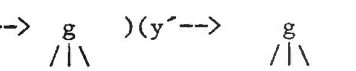


se méta-R,E-surréduit à l'occurrence 2

avec la règle f(f(y',x'),-x')--->y'

et le méta-E-unificateur

$(z \to z_2)(z_1 \to$  $)$

$(x' \to$  $)(y' \to$ $)$

en

qui sont méta-E-unifiables par

$(y \to z_3)(z_2 \to z_3)$

se R,E-surréduit à la même

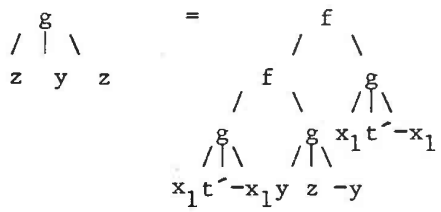
occurrence, avec la même règle

et par exemple le E-unificateur

$(x \to$ $)(x' \to$ $)(y' \to$ $)$

où t' ∈ $Mir_o(F,X)$.

en



se R,E-surréduit à la même

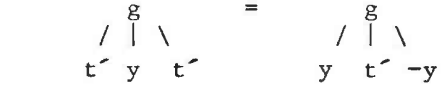
occurrence, avec la même règle

et le E-unificateur

$(x_1 \to y)(z \to t')(x' \to$ $)$

$(y' \to$ $)$

en



qui sont E-unifiables par

$(y \to t')$

7.3-AMELIORATION DANS LE CAS LINEAIRE

Il est intéressant de tester à chaque étape si l'équation obtenue est linéaire en une variable; auquel cas, il est inutile de poursuivre la branche correspondante dans l'arbre des R,E-surdérivations, car tout autre A-unificateur obtenu par la suite est supérieur à celui de l'équation linéaire.

En effet, si

$*(t,t) \text{ -V---}>_{[\sigma_0]} *(t_1,t'_1) \ldots \text{ -V---}>_{[\sigma_i]} *(t_i,t'_i) \ldots \text{ -V---}>_{[\sigma_{n-1}]} *(t_n,t'_n)$

avec $*(t_i,t'_i)$ linéaire et μ l'unificateur minimal de t_i et t'_i,

t_n et t'_n E-unifiables par μ', il est clair que $\mu'\sigma_{n-1}\ldots\sigma_i$ A-unifie

t_i et t'_i, donc $\mu \leq_A \mu'\sigma_{n-1}\ldots\sigma_i\ [V(t_i) \cup V(t'_i)]$,

et $\mu\sigma_{i-1}\ldots\sigma_0 \leq_A \mu'\sigma_{n-1}\ldots\sigma_0\ \ [V(t) \cup V(t')]$

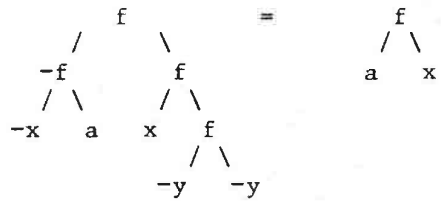
puisque $I(\sigma_{i-1}\ldots\sigma_0)$ est inclus dans $V(t_i) \cup V(t'_i)$.

Pour intégrer cette remarque au processus de méta-R,E-surréduction, il ne faut bien entendu tester la linéarité que par rapport à des variables de V et non par rapport aux méta-variables.

Cette amélioration permet d'éliminer certaines redondances, mais n'assure pas que l'ensemble de A-solutions trouvées soit minimal. Il est certainement possible d'introduire d'autres améliorations telles celles exposées par F.Fages [FAG,81] mais nous ne les développerons pas ici.

Pour terminer ce chapitre, nous allons donner un exemple de recherche de solutions d'une équation dans les arbres binaires signés où f est l'unique symbole de fonction d'arité 2, a une constante et x et y des variables. Nous ne représentons que deux branches significatives de l'arbre des R,E-surdérivations, afin de ne pas alourdir inutilement cet exemple. Dans ce cas précis, R,E-surréduction et méta-R,E-surréduction se confondent.

Nous n'indiquons à chaque étape que l'occurrence de la R,E-surréduction et la substitution R,E-surréductrice.

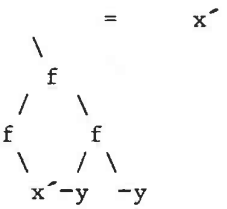


équation qui se R,E-surréduit

à l'occurrence 2 du deuxième terme

avec la substitution (x--> f)
/ \
-a x'

en

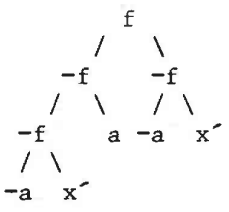


qui se R,E-surréduit

à l'occurrence 22 du premier terme

avec la substitution (y--> f)
/ \
-a x'

en



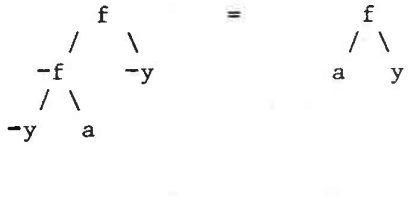
qui se R,E-réduit en -a = x'

Ces deux termes sont E-unifiables par (x'-->-a).

à l'occurrence 2 du premier terme

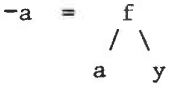
avec la substitution (x-->y)

en



qui se R,E-réduit

en



Cette équation est linéaire et a pour solution (y--> f).
/ \
-a -a

Nous obtenons ainsi deux fois la A-solution de l'équation de départ:

σ = (x--> f) (y--> f).
/ \ / \
-a -a -a -a

En essayant de R,E-surréduire aux autres occurrences basiques, on aboutit à un échec. { σ } est donc un ensemble complet de A-unificateurs des deux termes de départ, qui est de plus minimal dans ce cas.

CHAPITRE HUIT

APPLICATION DE LA THEORIE DES ARBRES SIGNES A L'INFERENCE DE SEQUENCE DE TERMES

Dans ce chapitre, nous allons donner un exemple d'utilisation de la théorie des arbres signés. Les questions suscitées par cette application sont nombreuses. Ce sera l'un des objectifs de ce chapitre de les poser.

8.1- INTRODUCTION

La théorie des arbres signés a été initialement introduite pour permettre d'étendre des méthodes développées en synthèse de programme à partir d'exemples ou en transformation de programmes. Les techniques connues jusqu'alors utilisaient soit la résolution des équations au sens du filtrage ou de l'unification, soit l'utilisation du processus de généralisation-filtrage décrit en particulier par R.S. Boyer, J.S. Moore, B. Wegbreit [B&M,75], [WEG,76], J.P. Jouannaud et Y.Kodratoff [J&K,81].

En synthèse de programme à partir d'exemples ou de traces, on cherchera par exemple à construire le programme qui, aux entrées

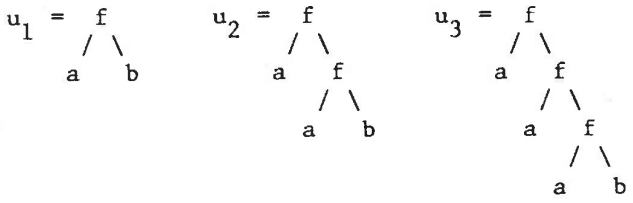
(a) , ((a) b) , (((a) b) c)

associe les sorties

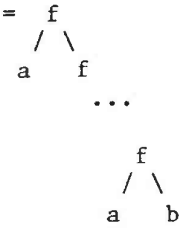
(a) , (a (b)) , (a (b (c))).

Le lecteur intéressé pourra trouver dans [KOD,79] un développement complet de cet exemple, et se référer aux travaux sur le sujet de A.W. Bierman, P.D. Summer, J.P. Jouannaud, Y. Kodratoff: [BIE,71], [SUM,75], [SUM,77], [J&K,81], [KOD,80].

On retrouve ce même probleme d'inférence si, par exemple, à partir de la suite finie suivante:



on cherche à inférer u_i =



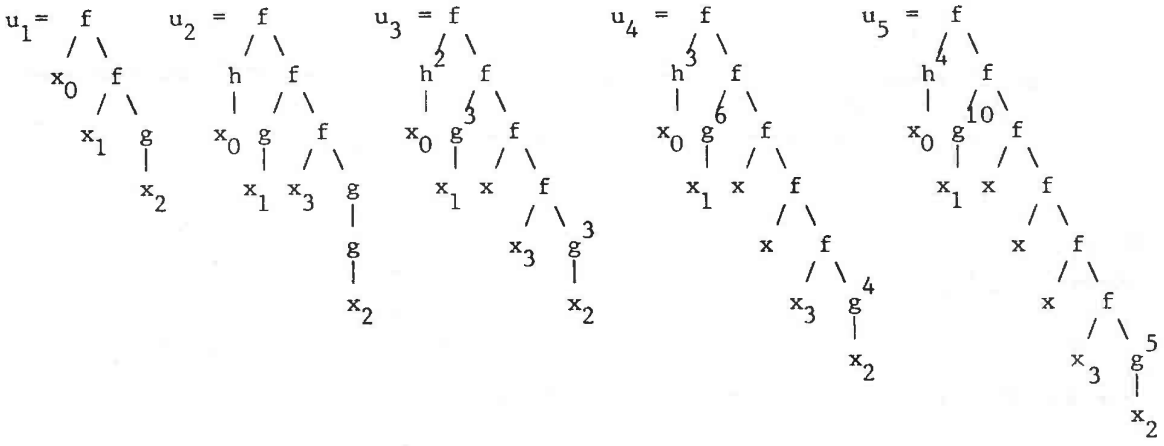
Une autre application récente de ce type d'inférence est constitué par la recherche de schéma de règles (ce que nous avons appelé méta-règle dans le chapitre deux), lorsque l'algorithme de Knuth et Bendix engendre une infinité de règles (la théorie des arbres signés en est un exemple). On peut, comme l'a suggéré Dershowitz [communication personnelle] et comme l'a montré Y. Kodratoff [KOD,82] essayer de fabriquer la méta-règle par inférence.
Notons que de telles séquences infinies de termes s'introduisent également, comme on a pu le voir dans cette thèse, au cours de processus d'unification. Plus généralement, chaque fois que l'on veut décrire un ensemble infini de termes par un mécanisme fini, en connaissant une partie finie de cet ensemble, ce genre de mécanisme d'inférence peut être utile.

Notons enfin que comme dans les méthodes rappelées ci-dessus, la détermination de la classe des programmes synthétisables en utilisant la théorie des arbres signés est un problème ouvert dans le cas général. Pour l'étude de cas particuliers, le lecteur intéressé pourra se référer à [J&K,79] ainsi qu'à [ANG,81]. Nous ne savons donc pas prouver que cette théorie apporte une

puissance d'expression strictement plus forte. Mais nous allons montrer sur un exemple la facilité d'expression et d'utilisation qui découle de l'utilisation des arbres signés.

8.2 UN EXEMPLE

Considérons la suite des termes suivants, où f et g sont des symboles d'arité 2 et 1 respectivement, et les x_i des symboles de variables.



Trouver une relation de récurrence sur la suite commencant par les termes u_1, u_2, u_3, u_4, u_5, c'est trouver une substitution permettant de calculer u_{i+1} à partir de u_i, ceci pour i=1,2,3,4. Si elle existe, on en déduit par inférence la relation entre deux éléments consécutifs de la suite permettant de construire de proche en proche tous les termes.

La méthode utilisée est le filtrage de u_i vers u_{i+1}: on essaie tout d'abord de filtrer au sens habituel, puis, en cas d'échec, on cherche un filtre dans les arbres signés.

Nous allons déterminer ci-dessous les différentes substitutions permettant de passer d'un terme de la suite au suivant, en cherchant les images de chacune des variables x_0, x_1, x_2, x_3.

* Remarquons tout d'abord que pour tous les termes de la suite:

$$\left(\begin{array}{l} x_0 \dashrightarrow h \\ \quad\quad | \\ \quad\quad x_0 \end{array}\right);$$ nous noterons cette substitution α,

et $(x_3 \dashrightarrow x)$ que nous noterons β .

* Pour la variable x_1 on a respectivement, pour passer de

u_1 à u_2 : $\alpha_1 = \left(\begin{array}{l} x_1 \dashrightarrow g \\ \quad\quad | \\ \quad\quad x_1 \end{array}\right)$ de u_2 à u_3 : $\alpha_2 = \left(\begin{array}{l} x_1 \dashrightarrow g^2 \\ \quad\quad | \\ \quad\quad x_1 \end{array}\right)$ etc...

Le filtrage n'est pas constant, mais il est immédiat d'en déduire que:

$$\alpha_{i+1}(x_1) = \left(\begin{array}{l} x_1 \dashrightarrow g \\ \quad\quad | \\ \quad\quad x_1 \end{array}\right) \alpha_i(x_1) \text{ pour } i=1,2,3,4.$$

Donc $\alpha_{i+1}(x_1) = \left(\begin{array}{l} x_1 \dashrightarrow g^i \\ \quad\quad | \\ \quad\quad x_1 \end{array}\right) \alpha_1(x_1) = \left(\begin{array}{l} x_1 \dashrightarrow g^{i+1} \\ \quad\quad | \\ \quad\quad x_1 \end{array}\right)$

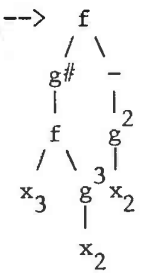
Ce que l'on suppose donc vrai pour i quelconque.

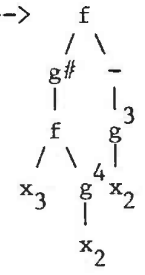
* Les substitutions δ_i permettant de filtrer u_i vers u_{i+1} et les δ_i ayant pour domaine $\{x_2\}$ ont respectivement comme image:

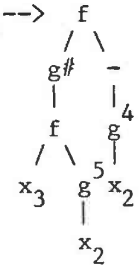
$t_1 = \begin{array}{l} g\# \\ | \\ f \\ / \ \backslash \\ x_3 \quad g^2 \\ \quad\quad | \\ \quad\quad x_2 \end{array}$ $t_2 = \begin{array}{l} (g\#)^2 \\ | \\ f \\ / \ \backslash \\ x_3 \quad g^3 \\ \quad\quad | \\ \quad\quad x_2 \end{array}$ $t_3 = \begin{array}{l} (g\#)^3 \\ | \\ f \\ / \ \backslash \\ x_3 \quad g^4 \\ \quad\quad | \\ \quad\quad x_2 \end{array}$ $t_4 = \begin{array}{l} (g\#)^4 \\ | \\ f \\ / \ \backslash \\ x_3 \quad g^5 \\ \quad\quad | \\ \quad\quad x_2 \end{array}$

Ici la relation n'est plus évidente à priori et nous allons à nouveau chercher une relation de récurrence entre t_i et t_{i+1}, c'est-à-dire entre $\delta_i(x_2)$ et $\delta_{i+1}(x_2)$.

On obtient comme séquence de filtres de t_i vers t_{i+1} de domaine $\{x_3\}$ les substitutions φ_i telles que:

$\varphi_1 : x_3 \longrightarrow$ 

$\varphi_2 : x_3 \longrightarrow$ 

$\varphi_3 : x_3 \longrightarrow$ 

Enfin pour tout i=1, 2, 3, 4, le filtre de $\varphi_i(x_3)$ vers $\varphi_{i+1}(x_3)$ est

$$(x_2 \longrightarrow \overset{|}{g}_{x_2})$$

Pour déterminer le filtre de t_i vers t_{i+1}, on obtient la relation de récurrence suivante:

$$\varphi_{i+1}(x_3) = (x_2 \longrightarrow \overset{|}{g}_{x_2}) \varphi_i(x_3) = (x_2 \longrightarrow \overset{|}{g^i}_{x_2}) \varphi_1(x_3)$$

Puis, pour déterminer la relation de récurrence entre les filtres δ_i, on peut écrire:

$$\delta_{i+1}(x_2) = t_{i+1} = \varphi_i(\delta_i(x_2)) = \ldots = \varphi_i \varphi_{i-1} \ldots \varphi_1 \delta_1(x_2)$$

$$\delta_{i+1}(x_2) = (x_2 \longrightarrow \overset{|}{g^{i-1}}_{x_2}) \varphi_1(x_2 \longrightarrow \overset{|}{g^{i-2}}_{x_2}) \varphi_1 \ldots (x_2 \longrightarrow \overset{|}{g}_{x_2}) \varphi_1 \varphi_1 \delta_1(x_2).$$

Par consequent un filtre σ_i de u_i vers u_{i+1} pour i=2,3,4 peut s'écrire:

$$\sigma_i = \delta_i + \alpha_i + \beta + \alpha$$

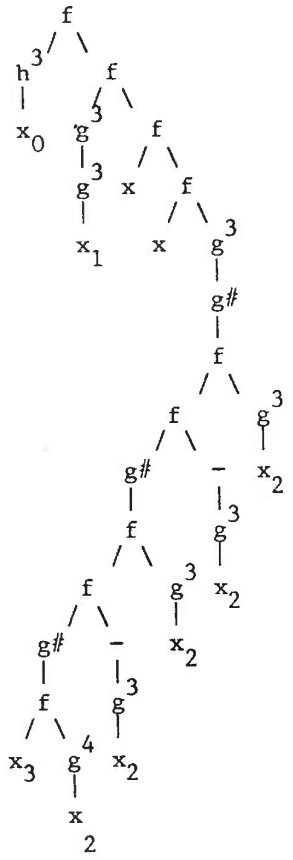
ce qui, compte tenu des relations trouvées plus haut, donne:

$$\sigma_i = \delta_i + (x_1 \longrightarrow \overset{|}{g^i}_{x_1}) + \alpha + \beta$$

Ce que l'on étend à i entier naturel supérieur à 1 et qui permet d'inférer la suite en entier.

Nous allons maintenant montrer sur un exemple comment la relation de récurrence obtenue permet d'obtenir une suite d'arbres signés dont les formes normales sont non signées.

$\sigma_3(u_3)$ =



La forme normale de ce terme est précisément u_4, comme on peut s'en convaincre facilement en effectuant les réductions.

Terminons en remarquant que l'on peut prouver dans ce cas que tous les termes de cette suite, obtenus par la relation de récurrence décrite ci-dessus, sont des éléments de l'algèbre des arbres signés dont la forme normale est un arbre non signé.

En général: le processus d'inférence d'une suite de termes, dont nous venons de donner un exemple, construit-il une suite de termes signés dont la forme normale (non signée) est la suite recherchée? Cette question est encore ouverte.

CONCLUSION

Il est possible de dégager de cette thèse plusieurs types d'apports:

* dans le domaine de la résolution d'équations dans les algèbres libres, puisque nous avons dégagé trois nouvelles méthodes d'unification:

-- l'une permise par l'introduction des algèbres signées et utilisant des transformations de type algébrique sur les équations.

-- une autre généralisant l'algorithme d'unification de Martelli et Montanari, basée sur une décomposition des équations en systèmes d'équations équivalents.

-- la troisième enfin donnant une méthode incrémentale de construction d'algorithmes d'unification fondée sur le processus de R,E-surréduction.

* dans les différentes manipulations d'ensembles infinis de termes, intervenant en tant qu'ensembles complets d'unificateurs ou dans des ensembles confluents de règles de réécriture. Dans les deux cas, sous le vocable "méta" (méta-règle, méta-unification), nous avons montré comment décrire finiment et manipuler ces ensembles infinis de termes. Il est clair que les méthodes utilisées sont générales et peuvent s'appliquer dans d'autres théories. La théorie des algèbres signées s'est donc révélée particulièrement fructueuse par les questions que son étude a suscitées.

* dans l'étude des systèmes de réécriture R pour lesquels l'algorithme de Knuth et Bendix diverge. Nous donnons un exemple de démarche qui parait s'avérer fructueuse: partant d'un système d'axiomes A, la localisation des axiomes provoquant la divergence permet de scinder A en R et E tel que

. R soit E-noethérien,

. E-commutant et

. un algorithme de E-unification est connu.

En appliquant le théorème de Peterson et Stickel, l'étude de la E-confluence se réduit à l'étude des paires E-critiques. On dispose en outre d'un algorithme de A-unification complet donné par le processus de R,E-surréduction.

Beaucoup de problèmes restent encore ouverts. Citons-en quelques-uns:

* Des améliorations sont peut-être à apporter en ce qui concerne par exemple la R,E-surréduction: peut-on affaiblir l'hypothèse de R,E-commutation que nous avons introduite?

* Les méta-règles, au lieu d'être uniquement un moyen de description d'un ensemble infini de règles, pourraient sans doute être utilisées dans un algorithme de complétion approprié. Pour cela, il faut en particulier savoir superposer des méta-règles et donc développer des techniques particulières.

* Un autre problème, concernant le phénomène de divergence de l'algorithme de complétion de Knuth et Bendix, est la détection d'ensembles infinis de règles schématisables par une méta-règle et la génération automatique de celle-ci. Nous avons vu dans le dernier chapitre que la théorie des arbres signés pouvait permettait d'inférer de facon simple de telles séquences de termes.

* Dans la théorie des arbres signés, l'apparition des méta-règles est liée à l'existence d'un sous-ensemble particulier d'axiomes E, dont on a étudié les propriétés. Peut-on en général relier l'existence d'une infinité de règles à un tel ensemble d'axiomes? Dans quelle mesure la méthode présentée dans ce travail se généralise-t-elle?

UN EDITEUR INTERACTIF DE TERMES SOUS FORME D'ARBRES

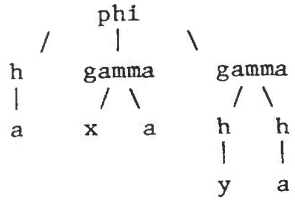
1- INTRODUCTION

Ce travail a été motivé par la constatation qu'un terme (c'est-à-dire un élément d'une F-algèbre libre engendrée par un ensemble de variables) se visualise et se manipule intellectuellement plus facilement sous la forme d'un arbre étiqueté que sous celle d'une formule linéaire parenthésée, que ce soit sous forme préfixée, postfixée ou autre.

L'éditeur permet donc de visualiser sur l'écran le terme

(phi (h (a)) (gamma x (a)) (gamma (h y) (h (a))))

où phi, gamma, h, a sont des symboles de fonctions d'arités respectives 3, 2, 1, 0, et x, y des variables, sous la forme



L'éditeur permet également de construire interactivement un terme, en le visualisant au fur et à mesure de sa construction, et de modifier un terme déjà existant.

L'ensemble est écrit en LISP, ce qui lui confère souplesse et portabilité d'une part, et lui permet d'autre part d'exploiter l'environnement de l'utilisateur. C'est ainsi que nous avons implémenté et visualisé par exemple les transformations d'équations permettant de calculer la solution minimale d'une équation linéaire dans l'ensemble des arbres signés.

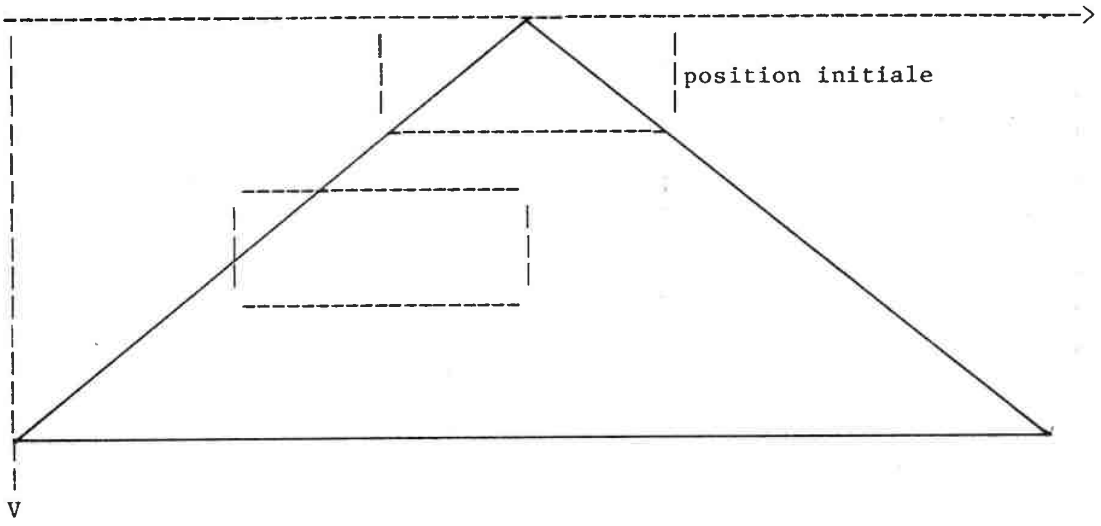
2- PRINCIPE DE L'EDITION

Nous allons décrire brièvement les structures de données utilisées ainsi que les principes suivant lesquels est construit l'éditeur.

2.1- LE REPERAGE

L'idée directrice est de considérer l'écran de visualisation comme une fenêtre que l'on déplace sur l'arbre, ceci afin de ne pas être limité par la taille de l'arbre.

On associe à l'arbre un repère qui permettra de calculer les coordonnées de chacun des noeuds ainsi que la position de la fenêtre (i.e. de l'écran), donc de connaitre à un instant donné l'ensemble des noeuds à visualiser.



2.2- LES STRUCTURES DE DONNEES

Nous manipulons trois représentations des arbres:

* La représentation sur l'écran, qui est la représentation sous forme d'un arbre étiqueté, dont nous avons donné un exemple plus haut.

* La représentation LISP du terme sous la forme d'une liste.

* La représentation graphique, dans laquelle on va placer l'information relative à la visualisation sur l'écran, de facon à ne pas la recalculer à chaque mouvement du curseur.

Par exemple au terme t dont la représentation sous forme de liste est

(f (a) (h x))

on associe le terme

$((f\ (x_f\ y_f)\ e_f\ occ_f)((a\ (x_a\ y_a)\ e_a\ occ_a))((h\ (x_h\ y_h)\ e_h\ occ_h)\ (x\ (x_x\ y_x)\ e_x\ occ_x)))$

obtenu en remplacant chaque symbole par la liste constituée de

- la liste de ses coordonnées dans le repère associé à l'arbre:
  par exemple $(x_f\ y_f)$ pour f.
- son encombrement, c'est-à-dire la largeur du sous-arbre dont le symbole est la tête: par exemple ici e_f pour f.
- son occurrence dans l'arbre: occ_f pour f.

3- DESCRIPTION DES COMMANDES DISPONIBLES

3.1 APPEL DE L'EDITEUR

On appelle l'éditeur en précisant en argument un identificateur que nous désignerons dans la suite par id. Si id est le nom d'un terme déjà connu, c'est ce terme qui sera visualisé, sinon cet identificateur servira à désigner le terme construit par l'éditeur au cours de cette session.

exemple: (ed bel-arbre)

* Si le terme est complet, la racine de sa représentation arborescente est visualisée (la fenêtre est positionnée dans la position initiale indiquée sur la figure ci-dessus) et l'éditeur attend un ordre de l'utilisateur.

* si le terme est incomplet (ceci résultant par exemple d'une session non terminée de l'editeur) et en particulier si on crée un terme, le curseur se positionne au premier noeud à compléter (la racine, si le terme est créé).

3.2 DEPLACEMENTS DU CURSEUR ET DE LA FENETRE

L'utilisateur peut déplacer le curseur dans toutes les directions sur l'écran. Si le curseur est au bord de l'écran, il "pousse" l'écran dans la direction demandée. Un contrôle est effectué, permettant de maintenir constamment la fenêtre sur une partie de l'arbre.

-> : déplace le curseur d'un caractère vers la droite

<- : même chose vers la gauche

↑ : vers le haut

↓ : vers le bas

shift ->: déplace le curseur de 80 caractères vers la droite

shift <-: déplace le curseur de 80 caractères vers la gauche

shift ↑ : déplace le curseur de 24 lignes vers le haut

shift ↓ : déplace le curseur de 24 lignes vers le bas

3.3- COMMANDES DE MODIFICATION

Toutes ces commandes sont préfixées par le caractère "ctrl"; si le curseur est mal placé, un commentaire est donné sur la 25eme ligne ou bien le terminal sonne.

ctrl C : Change le nom d'un symbole dans tout l'arbre. Le symbole à modifier et celui qui le remplacera sont demandés sur la 25eme ligne.

ctrl D : Détruit toute la branche de l'arbre à partir de l'endroit où se situe le curseur. Il est clair qu'il faut se situer à un noeud.

ctrl N : Permet de nommer un sous-arbre de l'arbre édité. Le sous-arbre dont la racine est pointée par le curseur aura dans la suite le nom donné.

ctrl R : Permet de changer l'identificateur du noeud pointé. Cet identificateur ne sera pas modifié ailleurs. Il faut être à un noeud.

ctrl S : Implémente la notion de substitution variable par variable. Un arbre donné déjà construit peut être substitué à la variable pointée par le curseur. Toutes les occurrences de la variable pointée seront instanciées. Le nom du sous-arbre à substituer est demandé sur la 25eme ligne.

3.4- POUR AVOIR DES INFORMATIONS

ctrl A : Donne l'arité d'un symbole, si celui-ci est connu.

ctrl H : Imprime une brève documentation en ligne.

ctrl O : Donne l'occurrence du noeud pointé par le curseur.

3.5- POUR SE RETROUVER SOUS LISP

ctrl E : Permet de sortir de l'éditeur. L'arbre édité est retourné en valeur sous sa forme LISP.

3.6- COMMANDES D'INSERTIONS

Elles seront toutes données lorsque le curseur est à un noeud. Dans ce cas

* l'éditeur accepte les chaines de caractères; elles se terminent par un "retour à la ligne" (CR)

* Si cette chaine est l'identificateur d'un symbole de fonction déjà connu, alors son arité étant connue, l'arbre est complété par le noeud et les branches vers les fils correspondant à son arité. Le curseur pointe alors sur le prochain noeud à compléter.

* Si cette chaine est l'identificateur d'un terme, alors sa représentation arborescente (complète ou non) est concaténée à l'arbre édité, à l'emplacement du noeud pointé par le curseur. Le curseur pointe ensuite sur le prochain noeud à compléter.
L'identificateur peut être le nom de l'arbre édité lui-même.

* Si l'identificateur est inconnu, son arité est demandée:

  -- les symboles de fonctions peuvent être d'arité quelconque.

  -- l'arité d'une variable sera désignée par: "v".

4- LA VERSION DISPONIBLE SUR H89 MUNI D'UNE INTERFACE GRAPHIQUE

L'éditeur tourne actuellement sur micro-ordinateur H89, 64K octets de mémoire centrale et pourvu d'une interface graphique (résolution 512x256). Le LISP utilisé est LELISP réalisé par J. Chailloux [CHA,79] qui fonctionne sous le système CP/M 2.2.

Des fonctions LISP permettant d'interfacer l'écran graphique (accès au point, tracé de segments) ont été écrites en assembleur Z80. Le lecteur intéressé par les algorithmes mis en oeuvre pourra consulter [BRE,65], [N&S,79], [SPR,81].

BIBLIOGRAPHIE

[ANG,81] ANGLUIN D.: "Summers revisited: Inference of term sequences"
Internal report. Yale university. To be publish (1981).

[A&N,80] ARNOLD A., NIVAT M.: "The metric space of infinite trees. Algebraic and topological properties"
Fundamenta Informaticae III.4, pp 181-205 (1980).

[BIE,78] BIERMANN A.W.: "The inference of regular LISP programs from examples"
IEEE Trans. syst. Cybern. 8, pp 585-600 (1978).

[BIR,35] BIRKHOFF G.: "On the structure of abstract algebras"
Proc. Cambridge Phil. Soc. 31, pp 433-454 (1935).

[BIR,67] BIRKHOFF G.: "Lattice theory"
3rd ed., A.M.S. Colloquium Publication no 25, Amer. Math. Soc., Providence R.I.(1967).

[BRE,65] BRESENHAM J.E.: "Algorithm for computer control of a digital plotter"
IBM System Journal, 4(1), pp 25-30, (1965).

[B&M,75] BOYER R. MOORE J.: "Proving theorems about LISP functions"
J.ACM 22, pp 129-144 (1975).

[B&M,77] BOYER R. MOORE J.: "A lemma driver automatic theorem prover for recursive function theory"
5th IJCAI, Boston (1977).

[CHA,79] CHAILLOUX J.: "VLISP 8.2 Manuel de référence"
Département d'informatique. Université de Paris VIII (1979).

[CHO,82] CHOQUE G.: "Etude et implémentation de la relation de plongement"
Rapport de DEA, Université de Nancy I.
Rapport interne CRIN 82-R-18 (1982).

[COUR,79] COURCELLE B.: "Arbres infinis et systèmes d'équations"
RAIRO Informatique Théorique, vol. 13-1, pp.31-48 (1979).

[COUR,79] COURCELLE B.: "Infinite trees in normal form and recursive equations having a unique solution."
Mathematical Systems Theory 13, pp 131-180 (1979).

[C&R,80] COURCELLE B. and RAOULT J.C.: "Completions of ordered magmas"
Fundamenta Informaticae III.1, pp 105-116 (1980).

[COUR,81] COURCELLE B.: "Fundamental properties of infinite trees"
Internationnal Summer School on Theoretical Foundations of Programming Methodology, Marktoberdorff. D.Reidel Publishing Company,Dordrecht, editor (1981).

[DER,79] DERSHOWITZ N.: "Orderings for term-rewriting systems"
Proc 20th Symposium on Foundations of Computer Science, pp 123-131 (1979).

[FAG,81] FAGES F.: Rapport de DEA,
Universite de Paris Sud Orsay (1981).

[FAY,78] FAY M.: "First order unification in an equationnal theory"
Master thesis. U. of California at Santa Cruz. report 78-5-002 (1978).

[FAY,79] FAY M.: "First order unification in equational theory"
Proc. 4th Workshop on Automata Deduction Texas (1979).

[H&H,80] HUET G. HULLOT J.M.: "Proofs by induction in equational theories with constructors"
Proc. 21th Symposium on Foundation of Computer Science (1980).

[H&O,80] HUET G. and OPPEN D.C.: "Equations and rewrite rules: a survey"
in "Formal Languages: Perspectives and open problems"
Ed. Book R., Academic Press.(1980)

[HUE,76] HUET G.: "Résolution d'équation dans les langages d'ordre 1,2...,ω"
Thèse d'état, Université Paris VII (1976).

[HUE,80] HUET G.: "Confluent reductions : abstract properties and applications to term rewriting systems"
J.ACM 27-4, pp 797-821 (1980).

[HUE,81] HUET G.: "A complete proof of the Knuth Bendix completion algorithm"
JCSS 23, pp 11-21 (1981).

[HUL,80] HULLOT J.M.: "Compilation de formes canoniques dans des théories équationnelles"
Thèse de 3eme cycle, Université Paris Sud Orsay (1980).

[J&K,79] JOUANNAUD J.P. KODRATOFF Y.: "Characterisation of a class of functions synthesised from examples by a SUMMERS like method using the BOYER-MOORE-WEGBREIT matching technique"
proc. 6th IJCAI Tokyo (1979).

[J&K,80] JOUANNAUD J.P. KODRATOFF Y.: "An automatic construction of LISP programs by transformation of functions synthesized from their input-output behaviour"
in P.A.I.S. number 2, MICHALSKI Editor (1980).

[K&J,81] JOUANNAUD J.P. KODRATOFF Y.: "Program synthesis from example of behaviour",
Proc. of the International Workshop on Program Construction. Chateau de Bonas. Ed. Biermann and Guiho. Reidel publish (1981).

[J&K,82] JOUANNAUD J.P., KIRCHNER H.: "Construction d'un plus petit ordre de simplification"
Soumis à RAIRO informatique théorique (1982).

[JLR,81] JOUANNAUD J.P., LESCANNE P., REINIG F.: "Recursive decomposition ordering"
in Formal description of programming concepts II
IFIP TC-2 conference. D.Bjørner Editor. North Holland (1982).

[JPK,80] JOUANNAUD J.P., PICARD M., KIRCHNER C. and H.: "Les arbres signés : un cadre algébrique pour la résolution d'équation dans les arbres"
Proc. Congrès AFCET-Informatique Nancy (1980).

[K&B,70] KNUTH D. BENDIX P.: "Simple word problems in universal algebras" in "Computational problems in abstract algebra" Leech J. ed. Pergamon Press (1970).

[KOD,79] KODRATOFF Y.: "A class of functions synthesized from a finite number of exemples and a LISP program scheme", Int. Journal of Computer and Inf. Sciences, Vol.8, No.6, pp 489-521 (1979).

[KOD,82] KODRATOFF Y.: "Synthèse de règles pour l'algorithme de Knuth et Bendix" Compte rendu des journées GROPLAN, GRECO-programmation. Bergerac (1982).

[KKJ,81] KIRCHNER C., KIRCHNER H., JOUANNAUD J.P.: "Algebraic manipulations as a unification and matching strategy for linear equations in signed binary trees." Proc. IJCAI 81 Vancouver (1981). Egalement rapport interne CRIN 81-R-029 (1981).

[K&K,81] KIRCHNER C., KIRCHNER H.: "Solving equations in the signed trees theory. Application to program derecursivation." Internal report CRIN 81-R-056 (1981).

[K&L,82] KAMIN, LEVY J.J.:"Two generalisations of recursive path ordering" Unpublished manuscript.

[KRU,60] KRUSKAL J.B. : "Well quasi ordering, the tree theorem and Vazsonyi's conjecture" Trans. Amer. Math.Soc. 95, pp 210-225 (1960).

[KOT,80] KOTT L.: "A system for proving equivalences of recursive programs" 5th Conference on Automated Deduction, Les Arcs (1980).

[KOT,80] KOTT L.: "Des substitutions dans les systèmes d'équations algébriques sur le magma. Application aux transformations de programmes et à leur correction." Thèse d'état, Université de Paris 7 (1980).

[K&P,80] KODRATOFF Y. and PAPON E.: "A system for program synthesis and program optimisation" Proc. AISB Meeting, Amsterdam (1980).

[LAN,79] LANKFORD D.S.: "A unification algorithm for abelian group theory" Report MTP-1. Math. dep., Louisiana Tech. U. (1979)

[L&B,79] LANKFORD D.S., BALLANTYNE A.M.: " The refutation completeness of blocked permutative narrowing and resolution" Proc. of the 4th Conference on Automata Deduction, Austin, pp 53-59 (1979)

[LES,79] LESCANNE P.: "Etude algébrique et relationnelle des types abstraits et de leur représentation" Thèse d'état, Université de Nancy I (1979).

[LES,81] LESCANNE P.: "A constructive proof of the well-foundedness of the recursive path ordering" Proc. Colloque AFCET "Les Mathématiques de l'Informatique" (1982).

[M&M,79] MARTELLI A. and MONTANARI U.: "An efficient unification algorithm" Technical report. Universite of Pisa (1979). A paraitre dans Transactions on Programming Languages and Systems.

[N&S,79] NEWMAN W.M. and SPROULL R.F.: "Principles of interactive computer graphics"
Second edition. McGraw-Hill Book Compagny (1979).

[NEW,42] NEWMAN M.H.A.: "On theories with a combinatorial definition of "equivalence"."
Ann. of Math. 43,2, pp 223-243 (1942)

[P&S,81] PETERSON G.E. and STICKEL M.E.: "Complete sets of reductions for equational theories with complete unification algorithms"
J.ACM 28, no.2, pp 233-264 (1981).

[PLA,78] PLAISTED D.: "A recursively defined ordering for proving terminaison of term rewriting systems"
Report 78-943, University of Illinois (1978).

[PLO,72] PLOTKING G.: "Building in equational theories"
Machine Intelligence 7, pp 73-90 (1972).

[P&W,78] PATERSON M.S. and WEGMAN M.N.: "Linear unification"
J. of Computer and Systems Sciences 16 pp 158-167 (1978).

[REI,81] REINIG F.: "L´ordre de décomposition: un outil incrémental pour prouver la terminaison finie de systèmes de réécriture de termes."
Thèse de 3eme cycle, Université de Nancy I (1981).

[ROB,65] ROBINSON J.A.: "A machine oriented logic based on the resolution principle"
J.ACM 12, pp 32-41 (1965).

[SIE,78] SIEKMANN J.: "Unification and matching problems"
Ph. D. Thesis, Memo CSM-4-78, U. of Essex (1978).

[S&S,81] SIEKMANN J. and SZABO P.: "Universal unification and regular equational ACFM theories"
Internal report. Universitat Karlsruhe (1981).

[SPR,81] SPROULL R.F.: "Using program transformations to derive line-draving algorithms"
Internal report CMU-CS-81-117. Carnegie Mellon University.
Pittsburg (1981).

[STI,81] STICKEL M.E.: "A unification algorithm for associative-commutative functions"
J.ACM 28, no.3, pp 423-434 (1981).

[SUM,75] SUMMERS P.D.: " Program construction from examples"
Ph. D. thesis, Yale University, New Haven, Connecticut (1975).

[SUM,77] SUMMERS P.D.: " A methodology for LISP program construction from examples",
J.ACM 24, pp 161-175 (1977).

[WEG,76] WEGBREIT: " Goal directed program transformation"
IEEE Trans. on Software Engineering 2, pp 69-80 (1976).

[W&H,81] WINSTON P.H. and HORN B.K.P. : "LISP"
Addison-Wesley Publishing Compagny (1981).

NOM DE L'ETUDIANT : Claude KIRCHNER

NATURE DE LA THESE : Doctorat 3ème cycle en Informatique

VU, APPROUVE

ET PERMIS D'IMPRIMER

NANCY, le 15 MARS 1982 .560.

LE PRESIDENT DE L'UNIVERSITE DE NANCY I,

l' Administrateur Provisoire

R. MAINARD

NOM DE L'ETUDIANT : Hélène KIRCHNER

NATURE DE LA THESE : Doctorat 3ème cycle en Informatique

VU, APPROUVE

ET PERMIS D'IMPRIMER

NANCY, le 15 MARS 1982 .559.

LE PRESIDENT DE L'UNIVERSITE DE NANCY I,

l' Administrateur Provisoire

R. MAINARD

RESUME

Nous nous intéressons dans cette thèse à différentes méthodes permettant de résoudre des équations dans l'ensemble des termes construits sur un ensemble de symboles de fonctions F et un ensemble de variables V.

La première méthode consiste à construire l'ensemble des arbres signés, qui est une extension de la F-algèbre libre engendrée par V, et le processus de résolution d'une équation linéaire utilise des transformations algébriques sur les équations, permises par l'introduction d'un nouveau symbole de fonction "-" et d'un ensemble d'axiomes A. Nous présentons en outre le concept important de méta-règle qui permet ici d'étudier le système d'axiomes A.

La seconde généralise la notion de surréduction à des théories équationnelles définies par un ensemble d'axiomes E et un système de réécriture R.

Une généralisation de l'algorithme d'unification de Martelli et Montanari permet de résoudre le problème de la E-unification.

L'application des deux méthodes précédentes à la théorie des arbres signés et la formalisation des notions de méta-E-unification et de méta-R,E-surréduction nous permet alors de résoudre des équations quelconques dans les arbres signés.

Nous présentons enfin un exemple d'application de la théorie des arbres signés à l'inférence de séquences de termes.

MOTS-CLES: réécriture, arbres, confluence, terminaison finie, méta-règles, théorie équationnelle, équations, multiéquations, unification, filtrage, surréduction, E-commutation, méta-E-unification, méta-R,E-surréduction, inférence.