DEMONSTRATION AUTOMATIQUE PAR DES TECHNIQUES DE REECRITURE

Thèse de Doctorat d'Etat en Mathématiques présentée par

Michaël Rusinowitch

soutenue le 6 novembre 1987 à Nancy 1.

devant la commission d'examen

Président:	Jean-Pierre JOUANNAUD Professeur
Rapporteurs:	Gérard HUET Professeur
	Jörg SIEKMANN Professeur
Examinateurs:	Jieh HSIANG Professeur
	Pierre LESCANNE Chargé de Recherche
	Pierre MARCHAND Professeur
	Jean-Luc REMY Chargé de Recherche

DEMONSTRATION AUTOMATIQUE PAR DES TECHNIQUES DE REECRITURE

Thèse de Doctorat d'Etat en Mathématiques présentée par

Michaël Rusinowitch

soutenue le 6 novembre 1987 à Nancy 1.

devant la commission d'examen

Président:	Jean-Pierre JOUANNAUD Professeur
Rapporteurs:	Gérard HUET Professeur
	Jörg SIEKMANN Professeur
Examinateurs:	Jieh HSIANG Professeur
	Pierre LESCANNE Chargé de Recherche
	Pierre MARCHAND Professeur
	Jean-Luc REMY Chargé de Recherche

REMERCIEMENTS

Je remercie

Jean-Pierre Jouannaud, pour son soutien constant durant l'élaboration de cette thèse, et la disponibilité dont il a fait preuve à mon égard. Ses analyses m'ont permis d'approfondir de nombreux points. Par son grand talent, je suis honoré qu'il préside ce jury.

Jieh Hsiang, pour les nombreux résultats qui sont les fruits de notre collaboration. Jieh m'a fait bénéficier de sa grande lucidité dans le domaine de la démonstration automatique, d'un rare esprit de synthèse et d'un véritable art de la présentation. Tant pour son amitié que pour ses qualités de chercheur, ce fût toujours pour moi un grand plaisir de travailler avec lui.

Gérard Huet, qui honore ce jury par son immense compétence dans le domaine du calcul formel. Ses travaux ont souvent guidé ma recherche, comme en témoignent les références.

Pierre Lescanne, mon directeur de thèse, qui a su me faire confiance et dont les connaissances et les conseils me furent extrêmement précieux, notamment dans le domaine de la terminaison.

Pierre Marchand et **Jean-Luc Rémy** qui me font l'amitié de participer au jury. Je les remercie pour leurs critiques pertinentes, leur patience à relire le manuscrit et pour les discussions que nous avons eues.

Jorge Siekmann, qui m'honore, par sa réputation internationale, de participer au jury .

Je remercie également

Emmanuel Kounalis pour sa contribution sympathique à certains résultats, et les nombreuses discussions qui l'ont accompagnée.

Jalel Mzali pour ses remarques judicieuses et les exemples intéressants qu'il m'a fournis

Ko Sakai pour sa participation à un chapitre.

Je remercie enfin les autres membres de l'équipe **EURECA** du CRIN pour l'amicale ambiance de travail qu'ils ont su créer et **Marie-Pascale Brouillard** pour son aide.

CONTRIBUTIONS

Les chapitres 2 et 3 de cette thèse ont été rédigés avec l'aide de Jieh Hsiang. Les résultats qu'ils contiennent ont, pour une partie, été présentés dans:
J. Hsiang, M. Rusinowitch, *A New Method for Establishing Refutational Completeness in Theorem Proving* , 8th CADE, Oxford, England, (1986).

Les résultats de la section 2 du chapitre 5 ont été obtenus en collaboration avec Jieh Hsiang et sont publiés dans :
J. Hsiang, M. Rusinowitch, *On Word Problems in Equational Theories*, 14th ICALP, (1987).

Les résultats de la section 3 du chapitre 5 ont été obtenus en collaboration avec Emmanuel Kounalis et sont présentés dans:
E.Kounalis, M.Rusinowitch, *On Word Problems in Horn Logic*, proceedings of the First Workshop on Conditional Term Rewriting, Orsay (1987).

Les résultats du chapitre 6 ont été obtenus en collaboration avec Jieh Hsiang et Ko Sakai et sont exposés dans:
J. Hsiang, M. Rusinowitch, K. Sakai, *Complete Set of Inference Rules for the Cancellation Laws* IJCAI 87', Milan, Italy , (August 1987).

INTRODUCTION 2
CHAPITRE 0 : 6
Introduction à la logique du premier ordre et aux systèmes de réécriture
1. Termes 7
2. Logique du premier ordre 10
3.Systèmes de Réécriture 12
CHAPITRE 1 : 16
Etude de quelques ordres de simplification
1. Preliminaires
2. Ordre des chemins de sous-termes.
3. Ordre récursif de décomposition.
CHAPITRE 2 : 34
Arbres sémantiques transfinis
1. Introduction 35
2. Préliminaires 36
3. E-Interprétations 40
4. Une nouvelle technique de preuve de complétude. 43
CHAPITRE 3 : 47
Stratégies de paramodulation
1. Stratégies de clauses ordonnées. 49
2. Relèvement des inférences. 54
3. Stratégies positives. 58
CHAPITRE 4 : 63
Complétude en présence de règles de réduction.
1. Complétude en présence de simplification and subsomption 65
2. Autres règles de réduction. 68
CHAPITRE 5 :
Stratégies de superposition. 70
1. Complétude de la stratégie de superposition. 72
2. Problèmes de mots dans les théories équationnelles 88
3. Problèmes de mots dans les théories de Horn. 95
CHAPITRE 6 :
Ensembles complets de règles d'inférence pour les axiomes de régularité. 107
1. Introduction 108
2. Règles d'inférence pour les axiomes de régularité. 109
3. Raffinements par les ordres de simplification. 112
4. Preuves de complétude des règles d'inférences 115
CONCLUSION 123

Cette thèse présente des méthodes de preuves automatique de théorèmes, favorisant l'usage des égalités comme règles de simplification, ainsi qu'une nouvelle technique pour prouver la complétude de ces méthodes.

Pour que la mécanisation du raisonnement ne demeure pas à jamais le rêve de Leibniz, la mise en code des lois logiques doit s'accompagner de stratégies pour guider la recherche des meilleures déductions. Le souci permanent d'orienter la production des conséquences vers le but a motivé la plupart des recherches en démonstration automatique. Celles-ci ont souvent dû leur succès à une étude fine de la structure des preuves.

Rappelons qu'un théorème admet par définition une preuve formelle et qu' un énoncé valide reste vrai, quelle que soit l'interprétation de ses symboles non logiques. Le théorème de complétude de Gödel établit l'équivalence de ces notions dans les théories du premier ordre. Mais les travaux de Church (1936) et Turing (1936) montrent qu'il n'existe pas d'algorithme qui puisse déterminer si une formule est un théorème. Néanmoins, il est possible de construire des procédures qui s'arrêtent avec succès uniquement sur les théorèmes: la validité est une propriété semi-décidable. En montrant qu'il suffit de tester la validité d'une formule dans une classe particulière de modèles, dits modèles canoniques, J.Herbrand (1930) a proposé la première méthode effective de démonstration automatique.

RESOLUTION

Pour démontrer une formule, on peut procéder par réfutation, c'est-à-dire montrer l'inconsistance de sa négation, en déduisant une formule contradictoire de cette dernière. J.A. Robinson (1965), s'appuyant sur les résultats de Herbrand, a construit une règle d'inférence appelée résolution, qui combine plusieurs règles classiques de la logique, et permet de prouver l'inconsistance des formules qui le sont effectivement. La règle de résolution restreint considérablement le nombre de conséquences à produire avant d'obtenir une réfutation : seules, sont engendrées les instanciations minimales permettant l'application immédiate de la règle de coupure, qui est une généralisation du *modus ponens*. Les instanciations minimales sont calculées au moyen de l'**algorithme d'unification.**

Les raffinements nombreux de la résolution permettent de restreindre encore l'ensemble des déductions possibles, comme la résolution sémantique (Robinson 1965b), la résolution linéaire (Loveland 1970; Luckam 1970), la lock-résolution (Boyer 1971), la résolution unitaire (Wos et al. 1964) et de contrôler l'ordre dans lequel elles sont construites, comme la SLD-résolution (Kowalski Kuhner 1971). Pour chaque raffinement se posent :

1. le problème de la correction: une formule réfutée est-elle effectivement inconsistante ?
2. le problème de la complétude: une contradiction est-elle dérivable de toute formule inconsistante ?

Ce dernier problème est l'objet essentiel de notre recherche. J.A. Robinson (1968) a prouvé la

complétude de la résolution par la méthode des arbres sémantiques, qui permet d'organiser de manière systématique l'ensemble généralement infini des interprétations de Herbrand. Cette méthode a été raffinée par R. Kowalski et P.J. Hayes (1970) pour s'appliquer à de nombreuses stratégies.

PARAMODULATION

La relation d'égalité intervient fréquemment dans les théories mathématiques. L'axiomatisation de cette relation conduit à la traiter uniformément comme les autres prédicats, sans considération du type de raisonnement concis, naturel et efficace qui lui est attaché, à savoir le remplacement d'un égal par un égal. Suivant les mêmes principes d'efficacité qui ont conduit à la résolution, L. Wos et G.A. Robinson ont introduit la règle de paramodulation, qui remplace plusieurs étapes de résolution par une instanciation suivie du remplacement d'un sous-terme. La paramodulation permet d'éviter l'utilisation de la plupart des axiomes d'égalité qui, sinon, produiraient de nombreuses formules intermédiaires inutiles. L.Wos et G.A.Robinson ont montré la complétude de la résolution associée à la paramodulation en présence des axiomes de réflexivité fonctionnelle. Brand (1975) a prouvé que toute réfutation peut éviter l'usage de ces axiomes. G.Peterson (1983) a formalisé la notion d' interprétation canonique égalitaire et généralisé la méthode des arbres sémantiques pour obtenir la complétude d'un raffinement essentiel de la paramodulation: il n'est jamais nécessaire de paramoduler dans une variable. Cette restriction supprime un nombre considérable d'inférences, puisqu'une variable peut s'unifier avec n'importe quel terme ne la contenant pas.

Dans cette thèse nous avons introduit une nouvelle technique pour prouver la complétude d'un ensemble de règles d'inférences. Cette méthode permet de raisonner sur des arbres sémantiques transfinis. Nous l'avons appliquée avec succès à plusieurs stratégies de paramodulation, comme la paramodulation ordonnée ou la paramodulation positive.

Ces stratégies, ainsi que toutes celles qui seront considérées plus loin,

- n'utilisent jamais les axiomes d'égalité (sauf x=x)
- n'appliquent jamais la paramodulation dans une variable

La notion essentielle qui sous-tend ces stratégies est celle d'ordre sur l'ensemble des termes et des formules: il s'agit de construire les conséquences logiques minimales des axiomes initiaux, avec pour but ultime de produire la plus petite formule, à savoir la clause contradictoire. Pour accélérer les preuves, en diminuant l'espace de recherche, toutes les stratégies considérées évitent de remplacer par paramodulation un terme par un terme plus grand (pour un ordre sur l'ensemble des termes donné en paramètre à la procédure). L'introduction des arbres sémantiques transfinis s'est justifiée pour pouvoir utiliser comme critère de comparaison des termes, les ordres de simplification, qui se sont avérés très puissants dans le contexte des systèmes de réécriture (Dershowitz 1982).

SIMPLIFICATION

Il est fondamental en démonstration automatique de maintenir l'information disponible sous la forme la plus compacte possible. Des règles d'inférence spéciales permettent de réduire la taille des formules traitées, à chaque étape du processus de déduction. La règle de subsomption permet de supprimer les formules redondantes. La règle d'élimination des tautologies écarte les trivialités, qui ne contribuent jamais à une preuve.

La règle de démodulation, qui consiste à utiliser les équations dans une seule direction pour réécrire les termes en des formes plus simples, est abondamment utilisée comme une heuristique très efficace (Wos, et al. 1967). En fait, l'influence de la démodulation sur la complétude des systèmes d'inférence est fort mal connue.

Nous avons démontré de manière très simple, par notre méthode de preuve, que la complétude de toutes les stratégies décrites dans ce travail est conservée en présence des règles de démodulation et de subsomption.

SUPERPOSITION

Cependant, dans le cadre plus restreint de la logique équationnelle, les travaux de Knuth et Bendix (1970) et ceux, nombreux, qu'ils ont suscités (Huet 1980) ont fourni des fondements théoriques à la règle de démodulation. Dans les théories qui sont présentées par des ensembles dits canoniques d'équations orientées, les termes sont égaux si on peut les simplifier jusqu'à obtenir des écritures identiques. Décider la validité dans ces théories est donc à la fois simple et efficace. La procédure de complétion de Knuth et Bendix permet de construire des systèmes canoniques à partir d'ensembles d'équations. Des catalogues de systèmes obtenus par cet algorithme sont présentés dans (Hullot 1980) et (Le Chenadec 1986).

Des auteurs ont remarqué depuis longtemps que la règle de superposition de l'algorithme de complétion n'est autre qu'une restriction de la paramodulation (Lankford 1975) (Brown 1974). Mais à la différence des stratégies de paramodulation, un problème essentiel de la procédure de complétion est celui de l'orientation, souvent impossible, des équations.

Par extension de la règle de superposition aux équations non orientables, nous avons pu obtenir un système d'inférence réfutationnellement complet pour la logique du premier ordre avec égalité. Lorsqu'on le restreint à traiter des ensembles formés uniquement d'équations, ce système fonctionne comme une procédure de complétion. Nous obtenons donc, en prime, une nouvelle preuve de correction de l'algorithme de Knuth et Bendix (sous réserve d'utiliser un ordre de simplification). Puisque l'orientation des équations importe seulement au moment de leur utilisation, lorsqu'elles sont instanciées, notre approche permet d'obtenir des systèmes canoniques contenant des équations non orientables. Elle s'applique également avec succès à la complétion des systèmes de réécriture conditionnels.

L'idée de remplacer certains axiomes par des règles d'inférence afin d'éviter la génération, par résolution, d'une kyrielle de formules non significatives, a été développée par Slagle (1972), notamment pour la théorie des ordres ou la théorie des ensembles. Plotkin (1973) a proposé de simuler des axiomes équationnels, comme l'associativité ou la commutativité, au moyen de nouveaux algorithmes d'unification. De nombreux travaux récents suivent la voie ouverte par Slagle, par exemple (Bledsoe et al. 1985) (Stickel 1985) (Manna et Waldinger 1986). Cependant les preuves de complétude, lorsqu'elles existent, sont toujours ardues. En adaptant la notion d'interprétation de Herbrand à d'autres théories axiomatiques que l'égalité, notre méthode fournit le cadre adéquat pour poursuivre le programme entrepris avec la paramodulation, à savoir la construction de systèmes de règles complets, remplaçant l'usage des axiomes. Ainsi, nous avons substitué aux axiomes de régularité (exprimant par exemple la possibilité de simplifier dans un groupe, un terme apparaissant de chaque côté d'une égalité) de nouvelles règles d'inférence, et prouvé leur complétude.

Plan de la thèse.

Le chapitre 0 introduit la logique du premier ordre ainsi que les systèmes de réécriture.

Le chapitre 1 est une étude comparative de certains ordres de simplification, utilisés en réécriture. Nous leur donnons des définitions plus simples et les étendons de manière à pouvoir comparer plus de termes.

Le chapitre 2 présente une nouvelle méthode pour prouver la complétude d'un ensemble de règles d'inférences. Après une définition constructive des interprétations de Herbrand égalitaires, nous définissons la notion d'arbre sémantique transfini et donnons un schéma de preuve de complétude.

Le chapitre 3 applique la méthode du chapitre précédent aux stratégies de paramodulation ordonnée et paramodulation positive.

Le chapitre 4 étend la technique du chapitre 2 au cas où l'on ajoute des règles de réduction comme la subsomption ou la simplification.

Le chapitre 5 introduit une stratégie complète pour la logique du premier ordre avec égalité, construite sur la règle de superposition. Restreint à des équations, ce système d'inférence généralise l'algorithme de Knuth et Bendix (sous réserve d'utiliser un ordre de simplification total sur les termes clos). Appliqué à des clauses de Horn, le système d'inférence est interprété comme une procédure de complétion de règles conditionnelles.

Le chapitre 6 propose une méthode générale pour remplacer des axiomes par des règles d'inférence complètes. Cette méthode est appliquée aux axiomes de régularité.

CHAPITRE 0

Introduction à la Logique du Premier ordre et aux Systèmes de Réécriture.

1. Termes

Nous présentons les objets formels de la logique du premier ordre.

Définition 1.1: Signature.

Soit F un ensemble fini ou dénombrable. Une *signature* α est une application de F dans N. Les éléments de F sont appelés *symboles de fonctions*. L'entier $\alpha(f)$ s'appelle *arité* de f ; lorsque $\alpha(f)$ est nul, on dit que f est une *constante*.

Définition 1.2: α- algèbre.

Etant donné une signature α , une *α-algèbre* A est une paire (M_A,F_A) formée d'un ensemble non vide M_A et d'une famille de fonctions F_A telle qu'à tout f de F on peut associer une fonction f_A de F_A

$$f_A : M_A{}^{\alpha(f)} \to M_A.$$

Définition 1.3: Termes.

Etant donné une signature α de domaine F et un ensemble dénombrable V d'éléments appelés *variables*, nous appelons ensemble de *termes* et notons $T(\alpha, v)$ la α -algèbre libre engendrée par V, c'est-à-dire le plus petit sous-ensemble de mots T sur le vocabulaire $F \cup V \cup \{ (,) \}$ tel que :

1. $V \subseteq T$
2. pour tout symbole $f \in F$ et toute suite $t_1, t_2,\ldots, t_{\alpha(f)}$ d'éléments de T, $f(t_1, t_2,\ldots, t_{\alpha(f)}) \in T$.

Par abus de notation on écrira parfois T(F,V) au lieu de $T(\alpha,V)$. L'ensemble des variables qui figurent dans un terme t sera dénoté par V(t). Un terme sans variable s'appelle un *terme clos*.. L'ensemble des termes clos de T(F, V) sera noté T(F). On supposera toujours qu'il est non vide. Afin de repérer les symboles figurant dans un terme, nous utilisons le système des occurrences.

Définition 1.4: Occurrence.

Soit N* l'ensemble des suites finies d'entiers naturels; la suite vide est notée Λ et l'opération de concaténation est notée par un point. L'ensemble des *occurrences* d'un terme t de $T(\alpha, V)$ est :

$$\text{occ}(t) := \begin{cases} \Lambda \text{ si } t \in V \\ \{\Lambda\} \cup \{i.p : I \le i \le n,\ p \in \text{occ}(t_i)\} \text{ si } t = f(t_1,\ldots,t_n) \end{cases}$$

Pour toute occurrence o dans occ(t), nous définissons :

a- le sous-terme de t à l'occurrence o noté t /o par :

$t/\wedge = t$

$f(t_1,\ldots,t_n)/i.o = t_i/o$

b- le terme issu de t en remplaçant son sous-terme à l'occurrence o par t', noté t[o← t'] par

$t[\wedge \leftarrow t'] = t'$

$f(t_1,\ldots, t_n)[i.o \leftarrow t'] = f(t_1,\ldots, t_i[o \leftarrow t'],\ldots,t_n)$

Définition 1.5 : Substitution

Une *substitution* σ est une application de V dans T(F, V), telle que $\sigma(x) = x$ sauf pour un ensemble fini de variables noté Dom (σ).

Les substitutions sont étendues en des endomorphismes de T(F,V) par les règles :

$$\sigma(f(t_1,\ldots, t_n)) = f(\sigma(t_1),\ldots, \sigma(t_n))$$

Le résultat de l'application d'une substitution σ à un terme t sera parfois noté tσ ; on aura :

$$t\sigma\rho = \rho(\sigma(t)).$$

Une substitution qui est une permutation des variables s'appelle un *renommage*. On utilisera souvent la propriété suivante sans la mentionner :

Propriété 1.6 : Soit s, $t \in T(F, V)$, $r \in occ(s)$ et σ une substitution.
Alors $\sigma(s[r \leftarrow t]) = \sigma(s)[r \leftarrow \sigma(t)]$.

La preuve procède par récurrence sur la longueur de r.

Définition 1.7: Instance, Filtre.

Soit s et $t \in T(F, V)$. On dit que t est une *instance* de s (ou s filtre t) s'il existe une substitution σ telle que $\sigma(s) = t$. La substitution σ s'appelle *filtre* de s vers t. La relation "est une instance", dénotée par ≤ est un pré-ordre sur T (F, V) appelé *pré-ordre de subsomption*..

Propriété 1.8: si $s \leq t$ et $t \leq s$ alors il existe un renomage σ tel que $t\sigma = s$. On dit alors que s et t sont des variantes.

Définissons la relation de *stricte subsomption* < par : $s < t$ si $s \leq t$ et $t \not\leq s$

Propriété 1.9: > est une ordre *noethérien* sur T (F, V). (il n'y a pas de chaine infinie $t_1 > t_2 > \ldots$).

La preuve de ces propriétés se trouve dans (Huet 76).

Définition 1.10: Unification

Deux termes s et t sont dits unifiables s'il existe une substitution σ, appelé unificateur, telle que

$$s\sigma = t\sigma.$$

Propriété 1.11: Si deux termes s et t sont unifiables, il existe un unificateur σ tel que pour tout autre unificateur θ, on peut trouver une substitution ρ vérifant: $\theta = \rho \circ \sigma$

σ s'appelle *unificateur principal* de s et t. Il est unique à un renomage près des variables. De nombreux algorithmes pour trouver un unificateur principal ont été proposés. (Robinson 65) (Huet 76) (Martelli Montanari 76).

2. Logique du premier ordre

Les faits élémentaires de la logique s'expriment par les notions d'atome et de littéral.

Définition 2.1: Atome, Littéral, Clause.

On considère un ensemble R de symboles de relations, une application d'arité $\beta : R \rightarrow N$ et une algèbre de termes T(F, V).

L'ensemble des *atomes* noté A (R, F, V) est

$$\{P(t_1, \ldots, t_n) : P \in R, n = \beta(P)\}$$

Un *littéral* est soit un atome (littéral positif), soit sa négation (littéral négatif). Une *clause* est une disjonction de littéraux. On identifiera une clause avec l'ensemble de ses littéraux. De même un ensemble de clauses sera considéré comme une conjonction de clauses, dont toutes les variables sont universellement quantifiées. La clause ne contenant aucun littéral est la *clause vide* . Elle est notée $\square$.

Les notions d'ensemble des variables, de substitutions, d'unificateur principal s'étendent aux atomes et aux littéraux. On supposera toujours que deux clauses distinctes ont leurs ensembles de variables disjoints.

Si C est une clause, et σ une substitution, $\sigma(C)$ est la clause

$$\{\sigma(L) : L \in C\}.$$

Remarquons que $\sigma(C)$ peut contenir moins de littéraux que C.

Pour chaque formule logique du premier ordre, on peut construire une formule simple, appelée forme standard de Skolem, qui est une conjonction de clauses et dont la consistance est équivalente à celle de la formule initiale. Une forme standard de Skolem peut s'obtenir par algorithme (voir par exemple (Loveland 1978) (Marchand 1986)). Cela justifie notre intérêt essentiel pour les ensembles de clauses.

Pour donner un sens à une formule du premier ordre, il faut l'interpréter comme une assertion dans une structure:

Définition 2.2: Interprétation, Assignation

Soit : une signature relationnelle $\beta : \Re \rightarrow N$
une signature fonctionnelle $\alpha : F \rightarrow N$
et un ensemble de variable V.

On appelle interprétation I la donnée :
d'un ensemble non vide D_I (*domaine* de l'interprétation)
pour chaque symbole R de $\Re$ d'une application $I(R) : D_I^{\beta(R)} \rightarrow \{V, F\}$
pour chaque symbole f de F d'une application $I(f) : D_I^{\beta(f)} \rightarrow D_I$.

Une *assignation* ν de I est une application $\nu : V \rightarrow D_I$ qu'on étend par morphisme à T (F, V) et A ($\Re$, F, V) par les règles suivantes:

$$\nu (f (t_1,..., t_n)) = I (f) (\nu (t_1),..., \nu (t_n))$$
$$\nu (R (t_1,..., t_n)) = I (R) (\nu (t_1),..., \nu (t_n))$$

Par conséquent ν associe à chaque atome une valeur de vérité. L'assignation ν peut s'étendre aux littéraux et aux clauses par les règles :

$$\nu (\neg A) = \neg \nu (A)$$
$$\nu (A \vee B) = \nu (A) \vee \nu (B)$$

Définition 2.3: Consistance
Un ensemble de clauses S est *consistant* (ou satisfaisable) s'il existe une interprétation I telle que, pour toute clause C de S et toute assignation ν de I, on a: $\nu (C) = V$.

En logique du premier ordre, une formule admet en général un nombre infini d'interprétations. On ne peut donc pas vérifier l'inconsistance d'une formule par évaluations successives de toutes les interprétations de la formule. Nous allons voir qu'il suffit de considérer des interprétations définies sur un domaine fixé, à savoir T (F).

Définition 2.4: Interprétation de Herbrand
Une interprétation de Herbrand est une interprétation I dont le domaine est T(F), et telle que pour tout symbole de fonction f d'arité n :

$$I (f) : \quad T (F)^n \rightarrow T (F)$$
$$(h_1,..., h_n) \rightarrow f(h_1,...,h_n)$$

(Aucune condition n'est imposée sur l'interprétation des symboles de relations).

Théorème 2.5: Un ensemble de clauses S est consistant si il existe une interprétation de Herbrand I, telle que pour $C \in S$ et toute assignation ν de I on a $\nu (C) = V$.

3. Systèmes de Réécriture

Lorsqu'une théorie est présentée par un ensemble d'équations, le problème de la validité d'une égalité dans cette théorie est semi-décidable par raisonnement équationnel: en substituant des termes par des termes égaux, on essaie de construire une suite allant d'un membre de l'égalité à l'autre. Cette méthode, justifiée par un théorème de Birkhoff (1935), est grossièrement inefficace. D'où l'introduction du raisonnement par réécriture, qui considère les équations comme des règles de simplification, utilisables dans une seule direction. Pour que cette méthode soit correcte, il faut que l'application des simplifications aboutisse toujours à une forme irréductible (propriété de terminaison), et que deux termes égaux dans la théorie puissent se réécrire en un même troisième (propriété de Church-Rosser). Cette dernière propriété n'est, en général, pas vérifiée par les équations qui présentent la théorie. Cependant, la procédure de complétion de Knuth et Bendix (1970) permet parfois de construire un système "de Church-Rosser", en ajoutant de nouvelles équations déduites des règles dont les membres gauches se superposent, et en orientant ces équations.

Avant de présenter formellement la réécriture, citons quelques unes de ses nombreuses applications: problème du mot en algèbre universelle (Evans 1951) (Bergman 1978) (Knuth et Bendix 1970), problème du mot dans les algèbres finiment présentées (Le Chenadec 1983), unification (Hullot 1980) (C. Kirchner 1986), preuves par induction (Huet et Hullot 1980) (Jouannaud et Kounalis 1986), programmation logique (Dershowitz 1985).

3.1. Propriétés abstraites de la réécriture

Nous nous intéressons aux congruences induites par des équations sur un ensemble de termes T(F,V).

Soit une relation binaire $\rightarrow$. Son inverse est notée $\leftarrow$.

Les symboles $\rightarrow^+$, $\rightarrow^*$, $\leftrightarrow$ dénotent sa fermeture transitive, réflexive-transitive, symétrique respectivement.

Principe d'induction noethérienne: soit $\rightarrow$ une relation noethérienne et P une propriété. Alors,

$$Si \ \forall x \ [\forall y : x \rightarrow y \Rightarrow P (y)] \Rightarrow P (x) \ \ alors \ \ \forall x \, P (x)$$

La relation $\rightarrow$ définie sur un ensemble de termes est *monotone* si $s \rightarrow t$ implique $u [s] \rightarrow u[t]$, pour tous les termes u, s et t. Elle est stable par substitution si $s \rightarrow t$ implique $s\sigma \rightarrow t\sigma$ pour toute substitution σ. Si $\rightarrow$ est un ordre noethérien, monotone et stable, on dit que $\rightarrow$ est un *ordre de réduction*.

Une *équation* est une paire de termes (s, t) écrite s = t. Pour tout ensemble d'équations, le symbole $\leftrightarrow_E$ représente la plus petite relation symétrique contenant E, stable et monotone. Ainsi s $\leftrightarrow_E$t si et seulement si s est de la forme C [ρ ← uσ] et t de la forme C [ρ ← vσ] où r est une occurrence de C, σ une substitution et soit (u, v), soit (v, u) appartient à E. Donc $\leftrightarrow_E^*$ est la plus petite congruence stable qui contient E. Une congruence est monotone par définition.

Un *règle de réécriture* est un couple de termes noté s → t tel que toute variable de t figure dans s également. Un *système de réécriture* est un ensemble de règles de réécriture.

La *relation de réduction* $\rightarrow_R$ est la plus petite relation stable et monotone qui contient R.

Ainsi, s $\rightarrow_R$t (s se réduit à t ou se réécrit en t) si s est de la forme C[ρ ← lσ] et t de la forme C[ρ ← rσ] pour une occurrence ρ d'un terme C, une substitution σ et une règle l→r ∈ R.

Un système de réécriture R est *Church-Rosser* si, pour tout terme s et t, s $\leftrightarrow_R^*$t implique qu'il existe un terme u tel que s $\rightarrow_R^*$ u $_R\leftarrow^*$t. Il termine si $\rightarrow_R$ est noethérien. Donc un système de réécriture termine si et seulement s'il est contenu dans un ordre de réduction. Un système de Church-Rosser qui termine est appelé canonique. Un terme est irréductible par R s'il n'existe aucun t' tel que t $\rightarrow_R$t'. Une forme normale de t pour $\rightarrow_R$ est un terme irréductible t' pour lequel t $\rightarrow_R^*$t'. Lorsque R est canonique, chaque terme t admet une unique forme normale pour $\rightarrow_R$ notée t $\downarrow_R$.

3.2. Complétion.

L'idée centrale des procédures de complétion est d'ajouter aux axiomes orientés en règles certaines conséquences critiques, obtenues par superposition des membres gauches des règles.

Définition 3.1: Paire critique (Knuth et Bendix 1970)
L'équation rσ = lσ[p←dσ] est une *paire critique* entre les règles l→r et g→d (dont les variables sont supposées distinctes), si σ est l'unificateur principal de g et d'un sous-terme non variable l /p de l.

On désigne par *CP(R)* l'ensemble des paires critiques entre des règles non nécessairement distinctes de R. Suivant la présentation de Bachmair, et al. (1986), nous décrivons la complétion comme un ensemble de règles d'inférence soumises à un contrôle équitable. Ces règles d'inférence ont pour objets des paires (E;R) constituées d'un ensemble d'équations E et d'un ensemble de règles de réécriture R. Nous supposons que > est un ordre de réduction.

Effacer	$(E \cup \{s=s\}; R) \vdash (E ; R)$
Orienter	$(E \cup \{s=t\}; R) \vdash (E; R \cup \{s \rightarrow t\})$ si s > t.
Déduire	$(E;R) \vdash (E \cup \{s=t\} ;R)$ si s=t appartient à CP(R).
Simplifier 1	$(E \cup \{s=t\}; R) \vdash (E \cup \{s=u\}; R)$ si $t \rightarrow_R u$.
Simplifier 2	$(E; R \cup \{s \rightarrow t\}) \vdash (E; R \cup \{s \rightarrow u\})$ si $t \rightarrow_R u$.
Simplifier 3	$(E; R \cup \{s \rightarrow t\}) \vdash (E \cup \{u=t\}; R)$ si $s \rightarrow_R u$.

Commentaires:
Effacer: supprime les équations triviales, dont la contribution à une preuve équationnelle est toujours évitable.
Orienter transforme une équation orientable en une règle de réécriture.
Simplifier: réduit les termes du système à l'aide des règles disponibles. Pour prouver la correction de la procédure de complétion, Bachmair et al. (1986) imposent certaines restrictions supplémentaires pour appliquer Simplifier3.

Une procédure de complétion est un programme qui, partant d'un ensemble fini d'équations E_0 et de règles R_0, et d'un ordre de réduction > contenant R_0, construit une dérivation suivant les règles d'inférences précédentes:

$$(E_0;R_0) \vdash (E_1;R_1) \vdash \ldots$$

La complétion réussit s'il existe une étape n où E_n est vide et R_n est un système canonique. En fait si toutes les paires critiques possibles ont été considérées, dès que E_n est vide on peut assurer que R_n est canonique.

Nous dirons qu'une dérivation est *équitable* si :

$$CP(\cup_{i\geq 0} \cap_{j\geq i} R_j) \subseteq \cup_{i\geq 0} E_i$$

Le résultat fondamental de Huet (1981) exprimant la correction de la procédure de Knuth et Bendix peut alors s'énoncer:

Théorème 3.2: l'ensemble final de règles d'une dérivation équitable est canonique (pour la théorie initiale $E_0 \cup R_0$) si l'ensemble final des équations est vide.

Exemple 3.3: L'exemple le plus fameux de système canonique obtenu par complétion est celui dérivé des axiomes de la théorie des groupes par Knuth et Bendix (1970):

axiomes: $e * x = x \quad x^{-1} * x = e \quad (x * y) * z = x * (y * z)$

Ces axiomes sont orientés de gauche à droite. Après une complétion, voici l'ensemble final R de règles de réécriture :

$e * x \to x$	$e^{-1} \to e$
$x^{-1} * x \to e$	$(x^{-1})^{-1} \to x$
$(x * y) * z \to x * (y * z)$	$x * x^{-1} \to e$
$x^{-1} * (x * y) \to y$	$x * (x^{-1} * y) \to y$
$x * e \to x$	$(x * y)^{-1} \to y^{-1} * x^{-1}$

Pour prouver, par exemple, que:

$$(x^{-1} * (x * y))^{-1} = (x^{-1} * y)^{-1} * x^{-1}$$

dans la théorie des groupes, il suffit de montrer que chacun des membres de l'égalité peut se réécrire par R en la forme normale y^{-1}.

CHAPITRE 1

Etude de Quelques Ordres de Simplification.

Nous souhaitons construire des systèmes d'inférence qui traitent, autant que possible, les équations qu'ils dérivent comme des règles de simplification. Celles-ci permettent de maintenir toutes les expressions sous forme réduite, sans perte d'information, limitant la taille des clauses engendrées par une procédure de preuve ou de complétion. L'orientation de ces règles est un problème délicat, car il est essentiel que les réductions successives d'un terme conduisent à des formes irréductibles. Il est également souhaitable de pouvoir orienter le plus grand nombre d'équations . Les critères basés uniquement sur la taille des termes sont en général trop grossiers.

Rappelons qu'un système de réécriture R termine s'il n'y a aucun terme t à l'origine d'une suite infinie de réductions: $t = t_1 \rightarrow_R t_2 \rightarrow_R \ldots$. La terminaison d'un système de réécriture est indécidable en général (Huet et Lankford 1976) . Puisqu'il est donc impossible de vérifier a posteriori qu'un système de règles termine, il est nécessaire de choisir des orientations prévenant la non-terminaison.

Pour qu'un ensemble de règles R termine, il suffit de trouver un ordre noethérien > sur l'ensemble des termes, contenant la relation de réécriture $\rightarrow_R$ (Manna et Ness 1970). Afin de pouvoir vérifier cette inclusion en examinant uniquement les règles de R, on s'intéresse aux ordres de réduction. Rappelons que ces ordre sont noethériens et vérifient les propriétés de

monotonie : $s>t$ implique $f(\ldots s\ldots)>f(\ldots t\ldots)$ pour tous les termes de T(F,V),
stabilité : $s>t$ implique $s\sigma>t\sigma$ pour tous les termes s, t et pour toute substitution σ.

Le résultat suivant qui localise le test de terminaison est dû à Lankford (1977).

Théorème: Le système de réécriture R termine si et seulement si il existe un ordre de réduction > tel que toute règle $l\rightarrow r$ de R vérifie $l>r$.

Dershowitz (1982) a remarqué qu'une relation d'ordre contenant le plongement est nécessairement noethérienne d'après le théorème de Kruskal. Il a également donné une condition suffisante très simple pour qu'un ordre monotone > contienne le plongement, à savoir la

propriété de sous-terme : $f(\ldots s\ldots)>s$ pour tous les termes de T(F,V).

Un ordre monotone possédant la propriété de sous-terme s'appelle ordre de simplification. Remarquons qu'un ordre monotone total et noethérien est toujours un ordre de simplification. Il existe plusieurs méthodes pour construire des ordres de simplification.

Méthode des interprétations.
Cette méthode consiste à construire un morphisme de la α-algèbre T(F) vers une α-algèbre ordonnée (W,>) telle que l'image $\tau(f)$ de chaque opérateur f de F soit une application strictement croissante en chacun de ses arguments, vérifiant de plus $\tau(f)(\ldots x\ldots) > x$ pour tout x de W. Par exemple, il est possible d'interpréter les symboles de F comme des fonctions polynômiales sur R (Dershowitz 1979, BenChérifa et Lescanne 1986).

Méthode de Knuth et Bendix.

Dans les chapitres suivants nous nous intéresserons à des ordres de simplification qui sont totaux sur les termes clos. La construction de Knuth et Bendix permet d'obtenir assez facilement de tels ordres. Remarquons d'abord que les ordres de simplification peuvent se généraliser aux pré-ordres (i.e. relations réflexives et transitives). Soit $\geq$ un pré-ordre sur T(F,V) satisfaisant les conditions suivantes:

monotonie : $s \geq t$ implique $f(\ldots s\ldots) \geq f(\ldots t\ldots)$ pour tous les termes de T(F,V),

propriété de sous-terme : $f(\ldots s\ldots) \geq s$ pour tous les termes de T(F,V).

Nous dirons que $\geq$ est un pré-ordre de simplification. Définissons $s > t$ par $s \geq t$ et non $t \geq s$. Alors si toute les règles d'un système de réécriture sont orientées par la partie stricte $>$ d'un pré-ordre de simplification stable , il termine (Dershowitz 1982).

Décrivons maintenant la construction de l'ordre de Knuth et Bendix (1970):

Nous noterons $n(x,t)$ le nombre d'occurrences de la variable x dans le terme t. Soit $>_F$ un ordre sur F l'ensemble des symboles de fonctions et $\geq$ un pré-ordre de simplification sur l'ensemble des termes. Notons $\approx$ l'intersection des relations $\geq$ et $\leq$. On suppose que $f(\ldots t\ldots) \approx t$ implique que f est unaire et maximal pour $>_F$. L'ordre de Knuth et Bendix $>_{kbo}$ est défini récursivement par :

$s=f(s_1,\ldots,s_m) >_{kbo} g(t_1,\ldots,t_n)=t$

si pour toute variable x, $n(x,s) \geq n(x,t)$ et

soit $s > t$

soit $s \approx t$ et $f >_F g$

soit $s \approx t$ et $f=g$ et $(s_1,\ldots,s_m) >_{kbo}^* (t_1,\ldots,t_n)$

avec $>_{kbo}^*$ dénotant l'extension lexicographique de $>_{kbo}$.

L'ordre $>_{kbo}$ est un ordre de simplification (Dershowitz 1982). La méthode originale de Knuth et Bendix utilise un ordre total $>_F$ sur F et attribue un poids strictement positif à chaque symbole de constante et un poids positif aux autres symboles fonctionnels. Dans ce cas, l'ordre $>_{kbo}$ obtenu est stable et total sur l'ensemble des termes clos.

Dans ce chapitre, nous étudierons des ordres noethériens sur les termes induits par une précédence et définis de manière récursive comme le RPO (Dershowitz 1982), le PSO (Plaisted 1978), le RDO (Jouannaud et al. 1984) et l'ordre décrit dans (Kapur et al. 1985).

J. Symbolic Computation (1987) 3, 117–131

Path of Subterms Ordering and Recursive Decomposition Ordering Revisited

MICHAEL RUSINOWITCH

Centre de Recherche en Informatique de Nancy, Campus Scientifique BP 239, 54506 *Vandœuvre Cedex, France*

The relationship between several simplification orderings is investigated: the path of subterms ordering, the recursive path ordering and the recursive decomposition ordering. The recursive decomposition ordering is improved in order to deal with more pairs of terms, and is made more efficient and easier to handle, by removing useless computations.

Introduction

Rewriting systems enable one to prove equalities in systems described by equations. Their principle is based on orienting axioms (and some of their consequences called critical pairs) so that they are always applied in the same direction. Oriented equations are called *rewrite rules* and applying them is called *rewriting*. Each term is associated with a *normal form*, that is, a term that cannot undergo more rewriting. Equality between two terms is then equivalent to identity of their normal forms, provided the system is confluent. See, for example, Huet & Oppen (1980).

To be able to compute normal forms, rewritings must terminate. This is usually demonstrated by exhibiting a well-founded ordering that contains the rewriting relation. Dershowitz (1982) has shown that the simplification orderings are well founded.

In the following, we study four simplification orderings:

— the Path of Subterms Ordering (PSO) of Plaisted (1978);
— the Recursive Path Ordering (RPO) of Dershowitz (1979, 1982);
— the Recursive Decomposition Ordering (RDO) of Jouannaud *et al.* (1982);
— the Path Ordering (KNS) of Kapur *et al.* (1985).

All these orderings are defined by extending a basic ordering on function symbols, called a *precedence*.

In the first part of this work, we recall the definition of PSO and show that it contains RPO. Then, following the remarks of Dershowitz (1982, p. 293), we get, as a consequence, that PSO is well founded if and only if the precedence is well founded. A counterexample which shows that PSO is not included in RDO is given. By a slight change in the definition of PSO, we get a new ordering which possesses the following property called *incrementality* (see Jouannaud *et al.*, 1982): given two terms, the precedence can be automatically extended in order to orient them.

Our goal in the second part is to improve the implementation of RDO by eliminating many unnecessary comparisons. For example, comparing the terms $a(b(c))$ and $A(b(c))$ with the precedence $a < A$, generates four checks of "$a < A$". The idea is to first simplify

0747-7171/87/010117+15 $03.00/0

the terms by their common suffix, then compare the left parts of these terms: the result is now obtained with a single use of "$a < A$". The same remark is true for KNS. Based on this idea, a simplified version of RDO is proposed, one that does not require the fourth component of the elementary decompositions (called the "context" in Jouannaud *et al.*, 1982). We prove that the ordering thus obtained is equivalent to RDO. But this definition of RDO generates less computations than the previous ones. The section ends with an extension of RDO which is proven equivalent to KNS.

1. Preliminaries

1.1. MULTISET ORDERINGS

A *multiset* is a set with possibly repeated elements (see Dershowitz & Manna, 1979, or Jouannaud & Lescanne, 1982 for a full description). More formally, a multiset M is a mapping $E \to N$, where E is a set and N is the set of natural numbers. The set of all the multisets on E with finite carrier is denoted by $M(E)$. For x in E, we say that $M(x)$ is the number of occurrences of x in M, and we write $x \in M$ instead of $M(x) > 0$. The sum of two multisets M and N is the multiset $M+N$ such that $M+N(x) = M(x)+N(x)$ for all $x \in E$. More generally, if $M_1, M_2, \ldots, M_k$ are multisets,

$$\left(\sum_{s=1}^{k} M_s\right)(x) = \sum_{s=1}^{k} M_s(x).$$

Each ordering on E can be extended to $M(E)$:

DEFINITION 1.1. Let $<E$ be an ordering on E. We can define an ordering on $M(E)$ by:

$$M1 \ll E\ M2 \text{ iff } M1 \neq M2$$

and

$$\forall x \in E\ (M2(x) < M1(x) \Rightarrow \exists\, y \in E\ x <E y \quad \text{and} \quad M1(y) < M2(y)).$$

If we denote the minimum of two mappings $M1$ and $M2$ by $M1 \cap M2$, we get a generalisation of the intersection of two sets. The next lemma states that, to compare multisets, we may first delete their common instances:

LEMMA 1.2.

$M1 \ll E\ M2$ *iff* $(M2 \neq \emptyset$ *and* $\forall x \in M1 - (M1 \cap M2)\ \exists\, y \in M2 - (M1 \cap M2)\ x <E y)$.

Inclusion of orderings is preserved when they are extended to multisets:

LEMMA 1.3. *The multiset extension of an ordering is monotonic, that is to say:*

$$(\forall x, y \in E\ x <E y \Rightarrow x <'E y) \Rightarrow (\forall M1, M2 \in M(E)\ M1 \ll E\ M2 \Rightarrow M1 \ll' E\ M2).$$

We can use this property to prove that well foundedness is preserved, too, by the multiset extension. Our proof does not use Konig's lemma as others usually do.

LEMMA 1.4. *$\ll E$ is well founded iff $<E$ is well founded.*

PROOF. $<E$ can be extended to a total well-founded ordering $<'E$; now, $\ll' E$ is well founded because it can be seen as a lexicographical ordering on ordered words on E. Since $\ll E$ is included in $\ll' E$ by Lemma 1.3, it is also well founded.

1.2. TERMS—OCCURRENCES

In the following, F denotes a finite set of function symbols, and ar is the arity of symbols in F. X is a set of variables. For any set E, E^* is the set of all finite sequences of elements of E. The empty sequence is denoted by ε. Concatenation of sequences is indicated by a dot:

$$(x_1, \ldots, x_k).(y_1, \ldots, y_l) = (x_1, \ldots, x_k, y_1, \ldots, y_l).$$

Thus, N^* is the set of all finite sequences of positive integers. Let $T(F, X)$ be the set of terms, that is the set of functions:

$t : N^* \to FUX$ whose domain $occ(t)$ is finite and satisfies:

$$\begin{cases} \varepsilon \in occ(t) \\ u.i \in occ(t) \text{ iff } u \in occ(t) \text{ and } i \in [1, ar(t(u))]. \end{cases}$$

Let t/u be the subterm of t at occurrence u, $t(u)$ the function symbol at u in t and $|t| = card(occ(t))$ the size of t. $T(F)$ is the set of closed terms.

EXAMPLE. If $t = f(g(a), g(h(a, b, c)))$ and $u = 213$, then $t/u = c$.

1.3. SIMPLIFICATION ORDERINGS

Simplification orderings have been introduced by Dershowitz (1979, 1982).

DEFINITION 1.5. An ordering $<$ on $T(F, X)$ is a *simplification ordering* if it has the following properties for every function symbol f:

compatibility property: $t_1 < t_2 \Rightarrow f(\ldots, t_1, \ldots) < f(\ldots, t_2, \ldots)$

subterm property: $t < f(\ldots, t, \ldots)$.

Orienting rules from left to right according to a simplification ordering ensures that the rewriting process terminates:

THEOREM 1.6 (Dershowitz, 1979). *A rewriting system with finitely many symbols is terminating if there exists a simplification ordering $<$ such that for all substitutions s and for all rule $g \to d$, $s(d) < s(g)$.*

Now, we are going to study two particular simplification orderings: the path of subterms ordering (PSO) and the recursive decomposition ordering (RDO).

2. Path of Subterms Ordering

Rather than comparing two terms directly, PSO compares two data structures built up from these terms: their paths of subterms. A path of subterms is the sequence of subterms on a path from the root to a leaf.

2.1. PATHS OF SUBTERMS

DEFINITION 2.1. Path of subterms.

The multiset of paths of subterms of t is

$$\mathrm{SPATH}(t) = \begin{cases} \{t\} \text{ if } t \text{ is a constant or a variable} \\ \sum_{i=1}^{m} \{t\}.\mathrm{SPATH}(t_i) \text{ if } t = f(t_1, t_2, \ldots, t_m), \end{cases}$$

where

$$\{t\}.\mathrm{SPATH}(t_i) = \sum_{g \in \mathrm{SPATH}(t_i)} \{t.g\}.$$

We define also:

$$\mathrm{PSPATH}(t) = \begin{cases} \emptyset \text{ if } t \text{ is a constant or a variable} \\ \sum_{i=1}^{m} \mathrm{SPATH}(t_i) \text{ if } t = f(t_1, \ldots, t_m). \end{cases}$$

EXAMPLE:

$$t = f(a, a, g(c))$$
$$\mathrm{SPATH}(t) = \{(t, a), (t, a), (t, g(c), c)\}$$
$$\mathrm{PSPATH}(t) = \{(a), (a), (g(c), c)\}.$$

DEFINITION 2.2. Permutative congruence $\sim$.

$$f(s_1, \ldots, s_n) \sim g(t_1, \ldots, t_n) \text{ iff } f = g \quad \text{and} \quad s_i \sim t_{\pi(i)}$$

for some permutation π of $\{1, 2, \ldots, n\}$.

DEFINITION 2.3. Let α be a sequence of terms, then SUBSEQU(α) is the multiset of subsequences of α

$$\mathrm{SUBSEQU}(\alpha) = \begin{cases} \emptyset \text{ if } \alpha \text{ is empty} \\ \{t, \varepsilon\}.\mathrm{SUBSEQU}(\beta) = t.\mathrm{SUBSEQU}(\beta) + \mathrm{SUBSEQU}(\beta) \text{ if } \alpha = t.\beta. \end{cases}$$

REMARK 2.4. If α is a path of subterms of t, then SUBSEQU(α) is actually a set, since every subterm of the path is distinct.

DEFINITION 2.5. If $<$ is an ordering on terms, then $<_{\mathrm{lex}}$ is a lexicographic-like ordering on sequences of terms defined by:

$$s <_{\mathrm{lex}} t \text{ iff } s = \emptyset \quad \text{and} \quad t \neq \emptyset$$

or

$$s_1 < t_1$$

or

$$s_1 \sim t_1 \quad \text{and} \quad (s_2, \ldots, s_m) <_{\mathrm{lex}} (t_2, \ldots, t_n)$$

where

$$s = s_1.s_2 \ldots s_m \quad \text{and} \quad t = t_1.t_2 \ldots t_n.$$

We are going to use the lexicographic extension of an intermediate ordering $<_i$ on terms to get an ordering on paths of subterms, used in the definition of PSO.

2.2. THE PATH OF SUBTERMS ORDERING ON $T(F)$

DEFINITION 2.6. Let $<$ be a precedence on F, $<_{pso}$ is defined recursively by:

$$s <_{pso} t \text{ iff } \mathrm{SPATH}(s) \ll \mathrm{SPATH}(t),$$

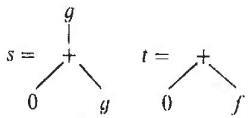
where <1 is an ordering on paths of subterms with:

$$a <1 b \text{ iff } \mathrm{SUBSEQU}(a) \ll_{ilex} \mathrm{SUBSEQU}(b),$$

where $<_i$ is an ordering on terms with:

$$u = f(u_1, \ldots, u_m) <_i v = g(v_1, \ldots, v_n) \text{ iff } f < g \text{ or } (f = g \text{ and } \mathrm{PSPATH}(u) \ll 1 \mathrm{PSPATH}(v)).$$

EXAMPLE 2.7.



with the precedence $0 < +$, $g < f$

$\mathrm{SPATH}(s) = \{(s, 0+g, 0), (s, 0+g, g)\}$ $\mathrm{SPATH}(t) = \{(t, 0), (t, f)\}$ $(s, 0+g, 0) <1 (t, f)$ because $s <_i f$, $0+g <_i t$, $0 <_i t$ $(s, 0+g, g) <1 (t, f)$ in a similar way and then $s <_{pso} t$.

We can prove this exactly as in Plaisted (1978):

PROPOSITION 2.8. *$<_{pso}$ is a simplification ordering.*

Moreover, $<_{pso}$ satisfies a compatibility property stronger than the one which is required in the definition of a simplification ordering:

PROPOSITION 2.9. *If*

$$\{s_1, \ldots, s_n\} \ll_{pso} \{t_1, \ldots, t_n\} \text{ then } f(s_1, \ldots, s_n) <_{pso} f(t_1, \ldots, t_n)$$

$<_{pso}$ is also monotonic with respect to the precedence:

PROPOSITION 2.10. *If $<$ is included (as a relation) in $<'$, then $<_{pso}$ is included in $<'_{pso}$.*

2.3. COMPARISON OF PSO AND RPO

Let us now recall the definition of RPO (cf. Dershowitz, 1979):

$$t = g(t_1, t_2, \ldots, t_n) <_{rpo} s = f(s_1, s_2, \ldots, s_m)$$

iff

$$(rpo1)\ f = g \text{ and } \{t_1, t_2, \ldots, t_n\} \ll_{rpo} \{s_1, s_2, \ldots, s_m\}$$

or

$$(rpo2)\ g <' \text{ and for all } t_i\colon t_i <_{rpo} s$$

or

$$(rpo3)\ not(g \leqslant f) \text{ and for some } s_i\colon t \leqslant_{rpo} s_i.$$

The main result of this section is:

THEOREM 2.11.

$$t <_{rpo} s \Rightarrow t <_{pso} s.$$

SKETCH OF THE PROOF. By induction on $|s|+|t|$. If $t <_{rpo} s$ by:

rpo1: then $\{t_1, t_2, \ldots\} \ll_{rpo} \{s_1, s_2, \ldots\}$, and by induction hypothesis $\{t_1, t_2, \ldots\} \ll_{pso} \{s_1, s_2, \ldots\}$; we can conclude $t <_{pso} s$ from $f = g$ and the generalised compatibility property of $<_{pso}$.

rpo2: then $g <_f$ and for all $t_i, t_i <_{rpo} s$. We have $\mathrm{SPATH}(s) \cap \mathrm{SPATH}(t_i) = \emptyset$, since $s \neq ti$. So, by induction, for each $a \in \mathrm{SPATH}(t_i)$, there exists $b \in \mathrm{SPATH}(s)$ with $a <_1 b$. Note that $t.a <_1 b$, by the following line of reasoning: $t.\mathrm{SUBSEQU}(a) \ll_{ilex} (s)$ since $t <_i s$ due to $g <_f$; '; $\mathrm{SUBSEQU}(a) \ll_{ilex} \mathrm{SUBSEQU}(b)$ because $a <_1 b$; so,

$$\mathrm{SUBSEQU}(t.a) = t.\mathrm{SUBSEQU}(a) + \mathrm{SUBSEQU}(a) \ll_{ilex} \mathrm{SUBSEQU}(b)$$

because (s) belongs to $\mathrm{SUBSEQU}(b)$, but not to $\mathrm{SUBSEQU}(t.a)$.

From this we get $\mathrm{SPATH}(t) \ll_1 \mathrm{SPATH}(s)$ and so, $t <_{pso} s$.

rpo3: then $t \leqslant_{rpo} s_i$; by induction $t \leqslant_{pso} s_i$; we conclude from the subterm property and the transitivity of $<_{pso}$.

COROLLARY 2.12. *If the precedence $<$ is total, then $<_{rpo}$ and $<_{pso}$ are the same ordering.*

PROOF. If $<$ is total, then $<_{rpo}$ is total on $T(F)/\sim$, and so is $<_{pso}$ from the last theorem. And, if $s \sim t$, s and t are neither comparable by $<_{rpo}$ nor by $<_{pso}$.

COROLLARY 2.13. *If $<$ is well founded, then $<_{pso}$ is well founded.*

PROOF. $<$ is included in a total well-founded ordering $<'$. It is known that $<'_{rpo}$ is well founded (cf. Dershowitz, 1979); but $<'_{rpo} = <'_{pso}$ and $<_{pso}$ is included in $<'_{pso}$ from the monotonicity property of PSO; the corollary follows as in Lemma 1.4.

REMARK 2.14. It will be shown in the next section that PSO is not included in RDO.

2.4. A VARIANT OF PSO

In most cases, the use of subsequences for comparing paths of subterms yields redundant computations. Considering paths of subterms as multisets provides a simple variant of PSO, with an additional property of incrementality.

DEFINITION 2.15. Let $<$ be a precedence. We define $<_{ps}$ recursively:

$$s <_{ps} t \text{ iff } \mathrm{SPATH}(s) \ll_2 \mathrm{SPATH}(t),$$

where $a <_2 b$ iff $\alpha \ll_j \beta$ (α and β denotes the multiset of subterms occurring in the path a and b, respectively) and

$$u <_j v \text{ iff } (f <_f g) \quad \text{or } (f = g \text{ and } \mathrm{PSPATH}(u) \ll_2 \mathrm{PSPATH}(v))$$

where $u = f(u_1, \ldots, u_n)$ and $v\, g(v_1, \ldots, v_m)$.

It can be proved that $<_{ps}$ is also a simplification ordering, which is monotonic, and contains RPO. The next example shows that $<_{ps}$ is also easy to use. Moreover, the precedence required to orient a rule can be easily computed. We conjecture that $<_{ps}$ and PSO are the same ordering.

EXAMPLE 2.16. Let us consider the rewriting system:

(1) $\quad \neg(\neg(x)) \to x$

(2) $\quad \neg(x \vee y) \to \neg(x) \wedge \neg(y)$

(3) $\quad \neg(x \wedge y) \to \neg(x) \vee \neg(y)$

(4) $\quad x \wedge (y \vee z) \to (x \wedge y) \vee (x \wedge z)$

(5) $\quad (y \vee z) \wedge x \to (y \wedge x) \vee (z \wedge x).$

An empty precedence is enough to orient (1), because of the subterm property of $<_{ps}$. To orient (2) we need to have:

$$\{\neg(x) \wedge \neg(y), \neg(x), x\} \ll_j \{\neg(x \vee y), x \vee y, x\},$$

but $\neg(x) <_j \neg(x \vee y)$.

Hence (2) is oriented with $<_{ps}$ iff $(\wedge < \neg)$ or $(\wedge < \vee)$. By exchanging the symbols '$\wedge$' and '$\vee$' we get the condition for (3): $(\vee < \neg)$ or $(\vee < \wedge)$. In order to have (4) directed with $<_{ps}$, we need to have:

$$\{(x \wedge y) \vee (x \wedge z), x \wedge y, x\} \ll_j \{x \wedge (y \vee z), x\},$$

but $(x \wedge y) <_j (x \wedge (y \vee z))$. To bound $(x \wedge y) \vee (x \wedge z)$ for $<_j$, the necessary and sufficient condition is $\vee < \wedge$. Then the other paths of the right-hand side of (4) are bounded too. Let us finally summarise the conditions $\vee < \wedge \sim \neg$.

3. RDO Revisited

3.1. DEFINITION OF RDO

A full description of RDO can be found in Jouannaud *et al.* (1982) or Lescanne (1982). The set $T(F, X, \Box)$ contains terms with at most one terminal occurrence of $\Box$, where $\Box$ is a symbol not in F that can be viewed as the empty term. If X is empty, we denote this set as $T(F, \Box)$. If u and v belong to N^*, then u/v is the word $w \in N^*$ such that $v.w = u$. Let t/u be the subterm of t at occurrence u, and $t[u \leftarrow t']$ the term obtained by replacing t/u by t' in t. If t belongs to $T(F, X, \Box)$

$$\|t\| = \|\{u \in occ(t);\ t(u) \neq \Box \text{ and } t(u) \notin X\}\|.$$

A *path* p of a term t is an occurrence such that $ar(t(p)) = 0$. Let p be a path of t, u a strict prefix of p, i the integer such $u.i$ is a prefix of p; $u.i$ is denoted by $suc(u, p)$.

DEFINITION 3.1. Elementary decomposition.

Given $t \in T(F, \Box)$, p, a path of t and u, a prefix of p, the *elementary decomposition* $d^p_u(t)$ *of t in u along the path p* is:

$\emptyset$ if $t(u) = \Box$; otherwise it is the quadruple: $\langle g, a, \psi, C \rangle$, where

$$g = t(u)$$
$$a = d^{p/suc(u,p)}(t/suc(u.p))$$

ψ is the multiset of other subterms of t/u, that is:

$$\{t/u.j : 1 \leqslant j \leqslant ar(t(u)) \text{ and } u.j \neq suc(u, p)\}$$
$$C = d^u(t[u \leftarrow \Box])\rangle,$$

where $d^p(t)$ is the *decomposition of t along the path p*, that is the set

$$\{d^p_u(t) : u \text{ is a prefix of } p\}.$$

We define also the multiset $d(t) = \{d^p(t) : p \text{ is a path of } t\}$.

REMARK. The term $t[u \leftarrow \square]$ is called the *context* of u in t. ψ is called the *neighbouring part*.

DEFINITION 3.2. RDO.

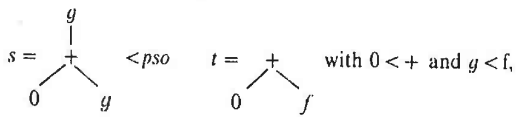
Given a partial ordering on F, we define the *recursive decomposition ordering* in the following way: $s <_{rdo} t$ iff $d(s) \ll_* \ll_* d(t)$, where $\ll_* \ll_*$ stands for the multiset of multisets ordering extending $<_*$, and $<_*$ is defined in the following way:

$$d_u^p(s) = \langle f, a, \phi, c\rangle <_* d_v^q(t) = \langle g, b, \psi, d\rangle$$

iff one of the following holds:

*dec*1: $f < g$
*dec*2: $f = g$ and $a \ll_* b$
*dec*3: $f = g$ and $a = b$ and $\phi \ll_{rdo} \psi$
*dec*4: $f = g$ and $a = b$ and $\phi = \psi$ and $c \ll_* d$.

REMARK. PSO is not included in RDO. As shown in a previous example:



whereas s and t are incomparable under the definition of $<_{rpo}$.

But we do not have $s <_{rdo} t$ because it is impossible to bound $d^{11}(s)$ with a decomposition along a path of t. A consequence is that PSO strictly contains RPO. The following example, given in Jouannaud *et al.* (1982) shows that RDO is not included in PSO:

We take an empty precedence on $F = \{f, a, b\}$,

$$s = f(f(a, b), f(a, b))$$

and

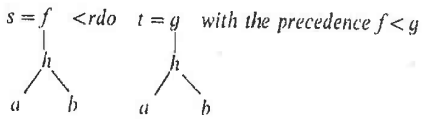
$$t = f(f(f(a, a), f(a, a)), f(f(b, b), f(b, b))).$$

It is not possible to compare s and t with PSO; nevertheless, we have $s <_{rdo} t$.

However, it is possible to prove that PSO, RPO and RDO are the same ordering when restricted to monadic terms.

3.2. REMOVING CONTEXTS FROM THE DEFINITION OF RDO

EXAMPLE. Suppose we want to show:



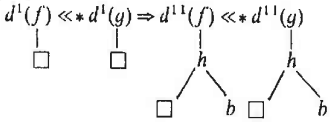
To get $d^{11}(s) \ll_* d^{11}(t)$ we have to prove:

$$d_{11}^{11}(s) <_* d_{11}^{11}(t) \text{ with } dec4$$

and

$$d_1^{11}(s) <_* d_1^{11}(t) \text{ with } dec4.$$

But this computation is useless, because the last inequality is, in fact, a consequence of the previous one. Indeed:



In the following, we give a new (equivalent) definition of RDO that gets rid of this redundancy, by eliminating all contexts from decompositions. Clearly, terms can be reconstructed from their decompositions with contexts, but they can be reconstructed without contexts as well. As a consequence, contexts are not really needed, as we show next.

DEFINITION OF RD.

DEFINITION 3.3. Simple decomposition.

Given $t \in T(F)$, p, a path of t and u, a prefix of p, the *simple decomposition* $D_u^p(t)$ *of* t *in* u *along the path* p is the triple $\langle g, a, \psi\rangle$, where

$$g = t(u)$$
$$a = D^{p/suc(u,p)}(t/suc(u, p))$$
$$\psi \text{ is the multiset } \{t/u.j : 1 \leqslant j \leqslant ar(t(u)), u.j \neq suc(u, p)\}$$

and the *simple decomposition of* t *along the path* p is the set

$$D^p(t) = \{D_u^p(t) : \text{is prefix of } p\}.$$

Let us also define the multiset: $D(t) = \{D^p(t) : p \text{ is path of } t\}$.

DEFINITION 3.4. RD.

The *simplified recursive decomposition ordering RD* is defined as follows:

$$s <_{rd} t \text{ iff } D(s) \ll_\bigcirc \ll_\bigcirc D(t)$$

with

$$D_u^p(s) = \langle f, a, \phi\rangle <_\bigcirc D_v^q(t) = \langle g, b, \psi\rangle$$

iff one of the following holds:

DEC1: $f < g$
DEC2: $f = g$ and $a \ll_\bigcirc b$
DEC3: $f = g$ and $a = b$ and $\phi \ll_{rd} \psi$.

REMARK 3.5. Note that we use a small d to denote elementary decompositions, while a capital D is used to denote simple decompositions.

RDO AND RD ARE EQUIVALENT.

If we need the contexts when comparing two paths p and q, using RDO, this implies that the two paths end with the same subsequence of subterms. Therefore, the last simple decompositions encountered on p are equal to the last simple decompositions of q. Hence, we do not need them when comparing p and q with RD, unlike RDO. This is the key aspect of our proof that RDO = RD. We state the previous remark more formally in the next lemma.

LEMMA 3.6. *Let p be a path of s, and q a path of t. Suppose that $D^p_u(s) = D^q_v(t)$, where u and v are prefixes of p and q, prespectively. Then:*

$$D^p(s) \ll\bigcirc D^q(t) \text{ iff } D^u(s[u\leftarrow\Box]) \ll\bigcirc D^v(t[v\leftarrow\Box]).$$

We can now state the main result.

THEOREM 3.7. *Given p and q, two paths of s and t respectively, we have:*

$$d^p(s) \ll* d^q(t) \text{ iff } D^p(s) \ll\bigcirc D^q(t).$$

Sketch of the proof. by induction on $\|s\|+\|t\|$.

First case: the paths p and q end with the same subsequence of subterms. So we can simplify p and q by their common suffix and are brought back to compare two paths u and v of $s[u\leftarrow\Box]$ and $t[v\leftarrow\Box]$, respectively, with u and v prefix of p and q, respectively. The result follows from the induction hypothesis.

Second case: p and q do not have the same tail. Hence, we never need contexts when comparing the elementary decompositions of p with those of q. Therefore, we could just as well perform the comparison with simple decompositions.

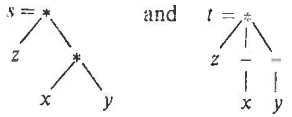
The last theorem yields immediately.

COROLLARY 3.8. *$s <rdo\ t$ iff $s <rd\ t$.*

When comparing two terms with RDO, we just need to compare the maximal decompositions of each term; this is the called $++$*strategy* in Jouannaud *et al.* (1982). This improvement is not possible with RD. However, Pierre Lescanne, who implemented both RD and RDO in his REVE system, noticed that, even with the $++$strategy, RDO appeared to be less efficient than RD, in most cases.

3.3. IMPROVING RDO

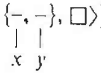
EXAMPLE 3.9. Given the terms:



and

with the precedence $* < -$, we can show that $s <rdo\ t$ is false. It is sufficient to prove that we have not $d^1(s) \ll* d^1(t)$. But

$$d^1(s) = \{\langle *, z, \{*\}, \Box\rangle\}, \quad d^1(t) = \{\langle *, z, \{-, -\}, \Box\rangle\}$$

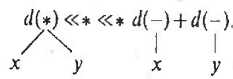


and

$$\{*\} \ll rdo\ \{-, -\} \text{ is false.}$$

However, we have:

$$d(*) \ll* \ll* d(-) + d(-).$$



On the other hand, we could verify: $s <pso\ t$ and $s <ps\ t$, so we have another counter-example showing that PSO is not included in RDO.

RDO fails to order s and t because it requires paths to be gathered in the neighbouring part of decompositions, so neither $-x$, nor $-y$ can bound $x*y$ and they cannot help each other to do that. More generally, if ϕ and ψ are two multisets of terms, then:

$$\phi \ll rdo\ \psi \text{ implies } \sum_{r\in\phi} d(r) \ll* \ll* \sum_{q\in\psi} d(q)$$

but the converse is false. So, if the last condition is taken instead of *dec*3 in the definition of RDO, the comparison should be more successful.

The previous example suggested to us an easy way to improve RDO. Instead of comparing the multisets of subterms that constitute the third part of decompositions, we compare the multiset sums of the paths of these subterms, without taking into account the original subterm they belong to.

In the following we shall extend RD in that direction. However, the same could be done with RDO itself.

DEFINITION 3.10. IRD.

We define on $T(F)$ the ordering IRD in the following way:

$$s <ird\ t \text{ iff } D(s) \ll\bullet \ll\bullet D(t),$$

where

$$D^p_u(s) = \langle f, a, \phi\rangle <\bullet D^q_v(t) = \langle g, b, \psi\rangle$$

iff one of the following holds:

DEC1: $f < g$
DEC2: $f = g$ and $a \ll\bullet b$
DEC3*b*: $f = g$ and $a = b$ and $\sum_{r\in\phi} D(r) \ll\bullet \ll\bullet \sum_{q\in\psi} D(q)$.

Now, with this slight change in the definition of RD, we get in Example 3.9: $s <ird\ t$, and we can prove straightforward that RDO is strictly included in IRD:

THEOREM 3.11. $s <rdo\ t \Rightarrow s <ird\ t$.

In Kapur *et al.* (1985), an ordering which we call KNS is described, and the authors give the following example of two terms that can be compared with their ordering, but for which RDO does not apply. This example is quite similar to Example 3.9.:

$$s = h(a(z), g(a(a(x)), x), g(b(b(y)), y))$$
$$t = h(a(z), g(a(x), b(y)), g(a(x), b(y))$$

and we may prove that $t <ird\ s$.

As a matter of fact, it will be shown in the next section that IRD and KNS are the same ordering. Now, if we compute the complexity of comparing two terms s and t using the definition of IRD, by the method described in Kapur *et al.* (1985), we get an upper bound $0(|s|^4 * |t|^4)$. With KNS, the upper bound is $0(|s|^5 * |t|^5)$. This is not surprising, since this last ordering uses contexts.

3.4. IRD AND KNS ARE EQUIVALENT

We proceed as follows:

(A) We give an alternative definition of KNS, which we think is simpler and more efficient.
(B) We show that IRD and KNS are equivalent.

(A) ALTERNATIVE DEFINITION OF KNS

In this subsection, the notations and definitions are taken from Kapur *et al.* (1985). We suppose, for simplicity, that no variable is involved.

DEFINITION 3.12. A *K-path* is a sequence of two-tuples, with the following properties:

Let $P = \langle f_1, T_1\rangle \ldots \langle f_n, T_n\rangle$ be a K-path. Then

1. f_i is the top level symbol of T_i for $1 \leqslant i \leqslant n$.
2. T_{i+1} is an immediate subterm of T_i.

With every couple $\langle f_i, T_i\rangle$ in the K-path P we can associate a *left-context* (LC) and a *right-context* (RC) defined as:

$$\mathrm{RC}(\langle f_i, T_i\rangle, P) = \begin{cases} \varepsilon \text{ if } i = n \\ \langle f_{i+1}, T_{i+1}\rangle \ldots \langle f_n, T_n\rangle \text{ if } i < n \end{cases}$$

$$\mathrm{LC}(\langle f_i, T_i\rangle, P) = \begin{cases} \varepsilon \text{ if } i = 1 \\ \langle f_1, T_1\rangle \ldots \langle f_{i-1}, T_{i-1}\rangle \text{ if } i > 1. \end{cases}$$

The K-path P is a *full K-path of a term t* if $T_1 = t$ and T_n is a constant. The multiset of the full K-paths of a term t is denoted by $\mathrm{MP}(t)$. The KNS ordering is defined by $s <_{kns} t$ if $\mathrm{MP}(s) \ll_p \mathrm{MP}(t)$, where $<_p$ is an ordering on the K-paths and is defined above:

Let $P1 = \langle k_1, T_1\rangle . \langle k_2, T_2\rangle \ldots \langle k_m, T_m\rangle$ and $P2 = \langle h_1, S_1\rangle \ldots \langle h_n, S_n\rangle$ be two K-paths.

We shall say that $P1 = P2$ if $m = n$ and $k_i = h_i$ and $T_i \sim S_i$ for all $i \in \{1, \ldots, n\}$. The K-path comparison is performed as follows:

$P2 <_p P1$ iff for all $\langle h_j, S_j\rangle$ in $P2$ there is $\langle k_i, T_i\rangle$ in $P1$ such that:

a. $h_j < k_i$, or
b. $h_j = k_i$ and
   1. $\mathrm{RC}(\langle h_j, S_j\rangle, P2) <_p \mathrm{RC}(\langle k_i, T_i\rangle, P1)$ or
   2. $\mathrm{RC}(\langle h_j, S_j\rangle, P2) = \mathrm{RC}(\langle k_i, T_i\rangle, P1)$ and $S_j <_{kns} T_i$ or
   3. $\mathrm{RC}(\langle h_j, S_j\rangle, P2) = \mathrm{RC}(\langle k_i, T_i\rangle, P1)$ and $S_j = T_i$ and $\mathrm{LC}(\langle h_j, S_j\rangle, P2) <_p \mathrm{LC}(\langle k_i, T_i\rangle, P1)$.

Now let us have a closer look at condition 3. This condition plays the role of the "context comparison" in RDO. If $P1$ and $P2$ have no common suffix, then we never have to use the test b.3. We can always reduce the situation to the previous one, thanks to the next result, which states a converse of Lemma 5 of Kapur *et al.* (1985).

LEMMA 3.13.: *Let P1, P2, P3, P4, P5 be K-paths such that* $P4 = P1 . P3$ *and* $P5 = P2 . P3$. *Then* $P5 <_p P4$ *iff* $P2 <_p P1$.

This leads to a more efficient definition of $<_p$ and the KNS ordering.

PROPOSITION 3.14. *Let P1' and P2' be the K-paths we get from P1 and P2 by deleting their common suffix. Then* $P2 <_p P1$ *iff for all* $\langle h_j, S_j\rangle$ *in P2' there exists* $\langle k_i, T_i\rangle$ *in P1' such that a. or b.1. or b.2. is true.*

We are now ready to prove:

(B) IRD AND KNS ARE EQUIVALENT

It is sufficient to prove that the path comparisons using $\ll_\bullet$ or $<_p$ always yield the same result.

PROPOSITION 3.15. *Let P and Q be two full K-paths of s and t, respectively. Then* $P <_p Q$ *iff* $D^p(s) \ll_\bullet D^q(t)$, *where* $p = u_1 \ldots u_n$ *and* $q = v_1 \ldots v_m$ *are paths of s and t, respectively, that verify:*

and

$$P = \langle s(\varepsilon), s\rangle . \langle s(u_1), s/u_1\rangle \ldots \langle s(p), s/p\rangle$$

$$Q = \langle t(\varepsilon), t\rangle . \langle t(v_1), t/v_1\rangle \ldots \langle t(q), s/q\rangle.$$

PROOF. (By induction on $|s| + |t|$) assume $P <_p Q$, and let P' and Q' be the K-paths we get by deleting the common suffix of P and Q. Let $\langle h_j, S_j\rangle$ be in P', with u such that $S_j = s/u$. We suppose that the $\langle k_i, T_i\rangle$ in Q' takes care of $\langle h_j, S_j\rangle$, and v is such that $T_i = t/v$. Let us show that $D^p_u(s) <_\bullet D^q_v(t)$. The only non-trivial case to consider is b.3, that is, $h_j = k_i$ and

$$\mathrm{RC}(\langle h_j, S_j\rangle, P) = \mathrm{RC}(\langle k_i, T_i\rangle, Q)$$

and $S_j <_{kns} T_i$. We have by definition of $\langle kns$: $\mathrm{MP}(S_j) \ll_p \mathrm{MP}(T_i)$. The relevant K-paths of this last inequality belong to smaller terms. Therefore, we can apply the induction hypothesis:

$$\sum_{a \in \mathrm{MP}(S_j)} D(a) \ll_\bullet \ll_\bullet \sum_{b \in \mathrm{MP}(T_i)} D(b).$$

The relation above is nothing else but DEC3b.

A similar argument proves the converse.

3.5. INCORPORATING STATUS IN RD

In the ordering RDOS defined by Lescanne (1984), when comparing two elementary decompositions having the same "leading symbol", the *status* of this symbol gives information about how to go on with the comparison: by comparing the "neighbouring part" as multisets or as lexicographically ordered sequences. The lexicographic statuses are very useful to orient associativity laws. They were first introduced by Kamin & Levy (1980) in order to extend RPO.

We suggest generalising the notion of status to include more strategies than the two,

multiset and lexicographic. We may, for instance, assign a "depth-first" or "breadth-first" status to some symbols according to how we handle the components of decompositions.

First, we modify slightly the definition of simple decompositions.

DEFINITION 3.16. Given p a path of t and u an occurrence of p, we define $D^p_u(t)$ as the triple $\langle f, a, \psi \rangle$, where

$$f = t(u)$$
$$a = D^{p/\mathrm{suc}(u,p)}$$
$$\psi \text{ is the } \textit{sequence } \langle t/u.j : 1 \leqslant j \leqslant ar(f) \rangle,$$

where $D^p(t)$ and $D(t)$ are defined as usual.

DEFINITION 3.17. Given an ordering $<_X$ on the elementary decompositions, the recursive decomposition ordering with status is defined in the following way:

$$s <_{rds} t \text{ iff } D(s) \ll_X \ll_X D(t).$$

The only property required for $<_X$ is:

$$t_i <_{rds} t'_i \text{ implies } \langle f, a, (t_1, \ldots, t_i, \ldots, t_n) \rangle <_X \langle f, a, (t_1, \ldots, t'_i, \ldots, t_m) \rangle.$$

Then, without any other condition, we can prove that $<_{rds}$ is a simplification ordering (see Kamin & Levy, 1980). Now let us give a good candidate for $<_X$:

DEFINITION 3.18.

$$D^p_u = \langle f, a, \phi \rangle <_X D^q_v = \langle g, b, \psi \rangle \text{ iff } f < g \text{ or } f = g$$

and case status of f is:

multiset-depth-first: (RDO-like)	$a \ll_X b$ or $a = b$ and $\sum_{S \in \phi} d(S) \ll_X \ll_X \sum_{T \in \psi} d(T)$
multiset-breadth-first:	$\sum_{S \in \phi} d(S) \ll_X \ll_X \sum_{T \in \psi} d(T)$
lexicographical-lr: (KAMIN&LEVY-like)	$\phi <_{lr} \psi$ where $<_{lr}$ is the left to right lexicographic extension of rds to sequences of term

and so on. . . .

We can add other cases as long as the property in Definition 3.17 is satisfied. By varying the choices of status for function symbols, RDS can take on features of the previously studied simplification orderings.

PROPOSITION 3.19.

1. *If all the statuses are multiset-depth-first, then* RDS = KNS = IRD, *but the time complexity of comparing s and t with* RDS *has an upper bound* $0(|s|^4 * |t|^4)$ *instead of* $0(|s|^5 * |t|^5)$.
2. *If the statuses are multiset depth first or lexicographic, then* RDS *properly contains* RPOS *and* RDOS, *respectively.*

REMARK. We do not know of any example that PSO can handle, but RDS cannot.

So, RDS is made more practical than others by its ability to include semantics for each function symbol by a generalised notion of status.

Conclusion

PSO and RDO have many common features: both of them work on paths of subterms, and extend RPO. Furthermore, PSO can be easily expressed in terms of decompositions. They essentially differ when comparing subterms with the same roots: PSO splits the paths and compares, in parallel, all paths of subterms issuing from the root; on the other hand, RDO goes on with the same path and checks the other paths later on. In other words, PSO works breadth first, and RDO works depth first. This is why these two orderings do not always give the same result. But we have been able to improve RDO by incorporating some ideas of PSO. We can sum up our results in the following diagram:

$$\begin{array}{ccccccc} \mathrm{PSO} & \supset & \mathrm{RPO} & \subset & \mathrm{RDO} = \mathrm{RD} & \subset & \mathrm{IRD} = \mathrm{KNS} \\ \cap & & & & & & \\ \mathrm{PS} & & & & & & \end{array}$$

An interesting feature of RD is its conceptual simplicity, which makes modifications easier. In particular, we can incorporate "status" à la Kamin & Levy (1980) in RD by comparing decompositions in a lexicographic way (instead of using multisets). Besides symbols with lexicographical status, one may have symbols with "depth-first" status or "breadth-first" status. So, according to the choice of the status of the function symbols, one can make RD more or less similar to either RDO or PSO.

I would like to thank Pierre Lescanne for his helpful guidance, and Jieh Hsiang, Jean-Pierre Jouannaud, Emmanuel Kounalis, Alain Quere and Jean-Luc Remy for reading the manuscript.

References

Dershowitz, N. (1979). *Orderings for Term Rewriting Systems*. Proc. 20th Symposium on Foundations of Computer Science.
Dershowitz, N. (1982). Orderings for term-rewriting systems. *Theor. Comp. Sci.* **17**, 279–301.
Dershowitz, N., Manna, Z. (1979). Proving termination with multiset orderings. *Commun. ACM* **22**, 465–476.
Huet, G., Oppen, D. C. (1980). Equations and rewrite rules: a survey. In: (Book, R., ed.) *Formal Language Theory: Perspectives and Open Problems*, pp. 349–405. London: Academic Press.
Jouannaud, J.-P., Lescanne, P., Reinig, F. (1982). Recursive decomposition ordering. In: (Bjormer, D., ed.) *Formal Description of Programming Concepts*, pp. 331–348. Amsterdam: North-Holland.
Jouannaud, J.-P., Lescanne, P. (1982). On multiset orderings. *Inf. Proc. Lett.* **15**, 57–63.
Kamin, S., Levy, J.-J. (1980). *Two Generalizations of the Recursive Path Orderings*. Unpublished note. Department of Computer Science, University of Illinois, Urbana, IL.
Kapur, D., Narendran, P., Sivakumar, G. (1985). A path ordering for proving termination of term rewriting systems. *CAAP 85*.
Lescanne, P. (1982). Some properties of decomposition orderings, a simplification ordering to prove termination of rewriting systems. *RAIRO Information theorique* **4**, 331–347.
Lescanne, P. (1984). Uniform termination of term rewriting systems—Recursive decomposition ordering with status. *CAAP 84*.
Plaisted, D. (1978). *A Recursively Defined Ordering for Proving Termination of Term Rewriting Systems*. University of Illinois, Sept. 78, UIUCDCS–R–78–943.

CHAPITRE 2

Arbres Sémantiques Transfinis.

Applications à la Preuve de Complétude des Systèmes d'Inférence.

Le problème de la complétude des stratégies de démonstration automatique est considéré comme délicat, dès que la relation d'égalité se trouve traitée de manière spécifique. Par exemple, la complétude de la résolution et de la paramodulation sans les axiomes de réflexivité fonctionnelle admet une preuve indirecte dans Brand (1975), comme corollaire de la complétude de la *méthode de modification*. Peterson (1983) en donne une preuve directe, s'appuyant sur un ordre de simplification isomorphe à ω sur l'univers de Herbrand. Cependant les ordres les plus utilisés pour orienter les équations en règles de réécriture sont transfinis. Les arbres sémantiques obtenus en ordonnant la base de Herbrand avec ces ordres sont donc, eux aussi, transfinis. Mais la méthode de Peterson, qui dérive de celle de Kowalski et Hayes (1970), repose sur la finitude des arbres sémantiques fermés: un nombre fini d'inférences doit permettre de réduire la *taille* d'un arbre jusqu'à zéro. La méthode de Peterson est donc inadaptée pour la plupart des stratégies traitant les équations de manière unidirectionnelle.

Nous proposons d'éviter ces difficultés en considérant "statiquement" l'ensemble (limite) de toutes les clauses inférées au cours d'une dérivation, et l'arbre sémantique fermé qui leur est associé. Pour une stratégie complète, cet arbre est vide chaque fois que l'ensemble initial des clauses est inconsistant. Partant de cette remarque, notre technique de preuve est la suivante: nous supposons que l'arbre associé à l'ensemble (limite) des clauses engendrées n'est pas vide; nous montrons que l'extrémité de sa branche droite est un noeud d'inférence, puis qu'une étape d'inférence permet de réfuter l'un de ses prédecesseurs, en contradiction avec l'hypothèse de non vacuité de l'arbre.

1. Introduction

Non seulement les ordres de simplification que nous utilisons permettent d'obtenir des preuves de complétude, mais ils servent aussi à raffiner les règles d'inférence de deux manières: premièrement, il suffit seulement de résoudre/paramoduler sur les littéraux maximaux; deuxièmement, lorsque l'on utilise une équation pour paramoduler, il suffit de considérer le plus grand des deux membres. Cette stratégie ordonnée, quand elle est restreinte au calcul des prédicats sans égalité est compatible avec la stratégie d'*ensemble support* (Wos Robinson 1968) (à l'inverse d'autres stratégies d'ordre/indexation). Elle peut s'étendre de manière naturelle au calcul des prédicats avec égalité. Et surtout, elle est compatible avec les règles d'inférence réductrices telles que la démodulation (simplification), la subsumption, et la règle d'élimination des tautologies.

Le plan de notre travail est le suivant: dans la Section 2, les notions préliminaires sont introduites ainsi que la définition des ordres de simplification complets. Ces ordres sont utilisés pour construire des interprétations égalitaires (E-interprétations) et des arbres sémantiques transfinis (arbres E-sémantiques) dans la Section 3. Dans la section 4 nous introduisons notre méthode de preuve fondée sur les arbres E-sémantiques. Nous appliquons cette méthode dans le Chapitre 3 pour prouver la complétude d'une *stratégie de clauses ordonnées* dans le cadre de la logique du premier ordre avec égalité. Nous présentons d'autres raffinements tels que les inferences bloquées et une étude comparative avec d'autres stratégies d'ordre/indexation. Comme autre exemple, nous introduisons la *stratégie positive*: cette stratégie combine la P1-strategie et la stratégie des clauses ordonnées. Nous en donnons la preuve de complétude. La complétude de la stratégie unitaire

positive pour les clauses de Horn avec égalité est un simple corollaire du résultat précédent.

Nous rappelons qu'aucune des stratégies envisagées dans cette thèse n'utilise les axiomes réflexifs fonctionnels ni la paramodulation dans les variables.

Notre méthode de preuve a également été utilisée pour construire une procédure de type Knuth-Bendix complète pour la réfutation (Hsiang Rusinowitch 1987a) qui est présentée dans le Chapitre 5. Elle peut s'étendre à un cadre plus général, pour des règles d'inférence construites sur d'autres axiomes. Le Chapitre 6 applique notre technique aux axiomes de régularité.

2. Préliminaires

Dans cette section, nous rappelons quelques notations. Nous supposons que l'ensemble des prédicats P contient un symbole binaire particulier: "=". L'ensemble des atomes sans variables (la *base de Herbrand)* est noté $A(P,F)$. Un *atome égalitaire* est un atome dont le symbole de prédicat est =. Nous supposerons toujours que = est commutatif. En d'autres termes, les atomes $s=t$ et $t=s$ sont considérés comme identiques.

2.1. Axiomes d'égalité.

La relation d'égalité est une relation de congruence; elle vérifie donc les axiomes suivants:

$\forall x\ (x=x)$.

$\forall x,y\ (x=y \supset y=x)$.

$\forall x,y,z\ (x=y \wedge y=z) \supset x=z)$.

Pour tout P, $\forall x,y\ (x=y \wedge P(x) \supset P(y))$.

Pour tout h, $\forall x,y\ (x=y \supset h(\ldots,x,\ldots)=h(\ldots,y,\ldots))$.

Cet ensemble d'axiomes sera noté K (d'après Chang Lee 1973). L'usage, sans restriction, de ces axiomes avec l'unique règle de résolution engendre une somme considérable de clauses non significatives, ou redondantes. Des efforts considérables ont été consacrés à construire des règles d'inférences permettant d'éviter l'axiomatisation de l'égalité. La règle de *paramodulation* (Wos Robinson 1968) en est l'exemple le plus notable.

2.2. Ordres sur les termes et les atomes.

2.2.1. Rappels sur les nombres ordinaux.

Avant d'introduire les ordres que nous servent à comparer les termes, nous rappelons quelques propriétés élémentaires des nombres ordinaux, qui seront utiles dans la suite.

Un ensemble bien ordonné est une paire $(A,>)$ où A est un ensemble et $>$ est un ordre bien fondé et total sur A. Deux ensembles ordonnés $(A,>_A)$ et $(B,>_B)$ sont *isomorphes* s'il existe une bijection h de A vers B telle que pour tout $a_1, a_2 \in A$, $a_1 \leq_A a_2$ si et seulement si $h(a_1) \leq_B h(a_2)$.

Un ensemble α est un *ordinal* si

(i) $<\alpha,\in>$ est un ensemble bien ordonné, et

(ii) $\beta \in \alpha$ implique $\beta \subseteq \alpha$.

Par définition l'ensemble vide, noté 0, est un ordinal. De même $1=\{0\}$, $2=\{0,1\}$, ..., et $\omega=\{0,1,2,\ldots\}$, $\omega+1=\omega\cup\{\omega\}$, etc... Un ordinal qui est fini (comme un nombre naturel) est un *ordinal fini*. Sinon (comme ω, $\omega+5$), c'est un *ordinal transfini*.

Théorème: *Tout ensemble bien ordonné* $(A,<)$ *est isomorphe à un unique ordinal.*

Ce nombre est appelé l'ordinal ou l'ordinalité de $(A,>)$. Soit α un ordinal, alors $\alpha^+=\{\eta:\eta\leq\alpha\}$ est le *successeur* de α, et α est le *predecesseur* de α^+. Le predecesseur d'un ordinal α sera noté α^-. Un ordinal non nul est un *ordinal limite* s'il n'est le successeur d'aucun ordinal. Autrement on dit que c'est un *ordinal successeur*. Par exemple, ω, 2ω, $\omega 2$, ω^ω, ... sont des ordinaux limites, tandis que 4, $\omega 3+11$ sont des ordinaux successeurs. Soit A un ordinal, sa *borne supérieure* est définie par $supA=\{\eta: \exists\xi\in A, \eta<\xi\}$. Par exemple, $sup5=4$. Soit α un ordinal, nous utiliserons les propriétés suivantes:

Lemme: (1) Si α est 0 ou un ordinal limite, alors $\sup\alpha = \alpha$.

(2) Si α est un ordinal successeur, alors $\sup\alpha = \alpha^-$.

L'arithmétique sur les ordinaux est définie comme une extension de l'arithmétique sur les nombres naturels:

$\alpha + 0 = \alpha$

$\alpha + \beta^+ = (\alpha + \beta)^+$

$\alpha + \gamma = \sup\{\alpha + \eta : \eta < \gamma\}$ où γ est un ordinal limite.

$\alpha \times 0 = 0$

$\alpha \times \beta^+ = \alpha \times \beta + \alpha$

$\alpha \times \gamma = \sup\{\alpha \times \eta : \eta < \gamma\}$ où γ est un ordinal limite.

Notons que $\alpha + 1 = \alpha^+ = \alpha \cup \{\alpha\}$. Une différence intéressante entre l'arithmétique ordinale et l'arithmétique naturelle est que la première n'est pas commutative.

Ainsi $1 + \omega = \omega \neq \omega + 1$.

2.2.2. Ordres de simplification complets.

Pour traiter l'égalité de manière efficace, nous proposons une construction par induction des modèles de Herbrand de K. Nous commençons par introduire des notions d'ordres sur les structures de termes et d'atomes.

Un ordre $<$ sur $T(F,X)\cup A(P,F,X)$ est un *ordre de simplification complet (CSO)* si c'est un ordre vérifiant:

O1: (*bonne fondation*):
< est bien fondé.
O2: (*linéarité sur les objets clos*)
< est total sur l'ensemble des objets clos: $T(F)\cup A(P,F)$
O3: (*stabilité*):
pour tout$w,v \in T(F,X)\cup A(P,F,X)$ et toute substitution θ, $w<v$ implique $w\theta<v\theta$.
O4: (*monotonie*):
pour tout $t, s \in T(F,X)$ *et* $w \in T(F,X)\cup A(P,F,X)$, $t<s$ implique $w[t]<w[s]$.
O5: (*propriété de sous-terme*):
pour tout $t,s,a,b \in T(F)$, où $t \leq s$, $a \leq b$ $u \in T(F)\cup A(P, F)$, et $w \in A(P, F)$,
1. si s est un sous-terme propre de u, alors $s<u$,
2. si s est un sous-terme de w et w n'est pas un atome égalitaire, alors $(s=t)<w$,
3. si s est un sous-terme propre de a ou b alors $(s=t)< (a=b)$.

Les conditions O1 and O2 permettent de construire les modèles de Herbrand par induction. La Condition O1 est en fait une conséquence de O2-O5 (Dershowitz 1982). Nous l'avons ajoutée aux autres pour insister sur son importance. La Condition O4 permet d'assurer que la valeur de vérité d'un atome est consistante avec les valeurs des atomes plus petits, et O5 force l'apparition d'un atome égalitaire avant tout atome qu'il peut réduire.

Une autre propriété importante d'un CSO est la suivante:

Proposition: Si $A(P,F)$ est infini et $>$ est un ordre de simplification complet, alors $(T(F)\cup A(P, F), >)$ est isomorphe à un ordinal limite.

La preuve, élémentaire, est laissée au lecteur.

Un CSO est une simple extension aux structures du premier ordre des ordres de simplification définis dans (Dershowitz 1982). Notre définition impose cependant quelques restrictions sur l'ordre entre un atome égalitaire et les autres atomes; la totalité sur les objets clos est également requise (Plaisted 1978). Une définition analogue a été proposée par (Peterson 1983). A la différence de celle-ci, la nôtre considère $s=t$ et $t=s$ comme le même atome. Nous avons seulement besoin de la totalité de < sur les objets clos $T(F) \cup A(P, F)$, alors que l'ordre de Peterson doit être isomorphe à ω, pour que les arbres sémantiques obtenus soient finis. Malheureusement, cette dernière condition l'empêche d'utiliser la plupart des ordres de simplification issus de la réécriture. Notre méthode, au contraire autorise n'importe quel ordre isomorphe à un ordinal sur $A(P, F)$. Ainsi, nous pouvons traiter l'ordre original de Knuth-Bendix (Knuth Bendix 1970), le *recursive path ordering* de (Dershowitz 1982), le *recursive decomposition ordering* de (Jouannaud et al. 1984), le *path of subterm ordering* de (Plaisted 1978). Pour un survol, consulter (Dershowitz 1985). En effet, à de légères adaptations près, la plupart d'entre eux vérifient cette propriété.

2.2.3. Exemples

Dans la suite nous donnons deux exemples. Dans le premier, les atomes sont comparés d'abord par leur symbole de prédicat puis par leurs arguments. Nous supposons que > est un ordre total sur les symboles de prédicats, = étant le plus petit. Nous supposons de plus que $>_f$ est un ordre de

simplification sur les termes qui est total sur les termes clos. Nous définissons $>_p$ sur $A(P,F,X)$ comme suit:

$P(s_1, \ldots, s_n) >_p Q(t_1, \ldots, t_m)$ si
- $P > Q$, ou bien
- $P=Q$, P est le prédicat d'égalité, et $\{s_1,s_2\} >>_f \{t_1,t_2\}$ où $>>_f$ est l'extension multi-ensemble de $>_f$, ou bien
- $P=Q$, P n'est pas le prédicat d'égalité, et $(s_1, \ldots, s_n) >_f (t_1, \ldots, t_m)$ sont comparés lexicographiquement

Intuitivement, pour $>_p$ tous les atomes égalitaires sont plus petits que les autres. Comme nous le verrons plus loin, de tels ordres limitent considérablement le nombre d'inférences.

Il est facile de vérifier que $>_p$ est effectivement un CSO. En général, il n'est pas isomorphe à ω. Supposons qu'il y a seulement deux symboles de prédicats = et P, une constante a, et deux fonctions unaires g et h. Supposons de plus que le *RPO* (Dershowitz 1982), avec $a < g < h$ sert à ordonner les termes de $T(F,X)$. Autrement dit, l'univers de Herbrand est ordonné de la manière suivante (rappelons que les atomes $s=t$ et $t=s$ sont considérés comme identiques).

$$a <_f ga <_f gga <_f ggga <_f \ldots <_f ha <_f gha <_f ggha <_f \ldots <_f hga <_f hgga <_f \ldots$$

Alors la base de Herbrand est ordonnée par $<_p$ de la manière suivante:

$$a=a <_p ga=a <_p ga=ga <_p gga=a <_p \ldots <_p ha=a <_p ha=ga <_p ha=gga <_p \ldots$$
$$\ldots <_p ha=ha <_p gha=a <_p \ldots <_p Pa <_p Pga <_p \ldots <_p Pha <_p Pgha <_p \ldots$$

Remarquons aussi que pour $>_p$, des atomes ayant des variables différentes peuvent être comparables. Par exemple,

$$P(y) >_p g(g(x)=x.$$

Malheureusement, il existe des ordres de simplification qui ne peuvent s'étendre en CSO. Considérons par exemple la signature $\{g,h,a,b\}$. Nous définissons un ordre de type RPO par:

$s>t$ si

- $s=g(a)$ et $t=g(b)$, *ou*
- $s=h(b)$ et $t=h(a)$, *ou*
- t est un sous-terme propre de s, *ou*
- $s=g(s_1)$, $t=g(t_1)$, et $s_1>t_1$, *ou*
- $s=h(s_1)$, $t=h(t_1)$, et $s_1>t_1$.

Il est impossible d'étendre > à a et b sans détruire la monotonie.

2.2.4. Incrémentalité des ordres.

La plupart des ordres utilisés en réécriture (e.g., recursive path ordering, recursive decomposition ordering, path of subterm orderings) sont de nature syntaxique. Voir (Dershowitz

1985) pour un résumé et (Rusinowitch 1986) pour une étude comparative. Ils sont construits à partir d'une *précédence* sur les opérateurs, et comparent les termes par leur structure syntaxique. Ils possèdent en général une propriété qui est très utile en démonstration automatique. C'est une propriété de monotonie par rapport à la précédence; nous l'appelons *incrémentalité*. Intuitivement, un ordre possède cette propriété, si, lorsque la signature est étendue, il est possible de construire une relation d'ordre (de simplification) contenant l'ancienne. Ainsi l'ordre n'a besoin d'être recalculé que sur les termes contenant de nouveaux symboles.

Pour apprécier l'importance de cette propriété, rappelons que l'introduction d'une formule dans un environnement, nécessite souvent l'addition de nouvelles constantes de Skolem.

3. E-Interprétations

Soit > un CSO sur la base de Herbrand $A(P,F)$ dont la suite des atomes est $\{A_i\}_{i<\lambda}$, λ étant son ordinalité. Etant donné un atome clos A_α, nous notons W_α le *segment initial* $\{A_i : i<\alpha\}$. Donc, W_α contient tous les atomes clos d'ordinalité inférieure à α, A_α étant exclus. Par exemple, $W_0 = \emptyset$, et $W_\lambda = A(P,F)$.

Une *E-interprétation partielle sur* W_α est une application I de W_α dans $\{V,F\}$ vérifiant:

- $I(s=s) = V$ si $s=s \in W_\alpha$
- si $(s=t), B[s], B[t] \in W_\alpha$ et $I(s=t) = V$, alors $I(B[s]) = I(B[t])$.

Une *E-interprétation* est une E-interprétation partielle définie sur W_λ, la base de Herbrand. Il est élémentaire de vérifier qu'une interprétation est une E-interprétation si et seulement si elle valide les axiomes d'égalité K.

Les E-interprétations peuvent s'étendre aux clauses sans variables de manière classique. Soit I une E-interpretation sur D, A un élément de D, et $C = L_1 \vee ... \vee L_n$ une clause sans variables dont tous les atomes sont dans D. Alors $I(\neg A) = \neg I(A)$ et $I(C) = I(L_1) \vee ... \vee I(L_n)$. Nous écrirons parfois $I \models C$ pour $I(C)=V$ et $I \not\models C$ pour $I(C)=F$. Etant donné une E-interprétation et une clause C, I *satisfait* C (ou C est valide dans I) si pour toute instance close C' de C, $I \models C'$. C est *E-consistante* si C est valide dans au moins une E-interprétation. Autrement elle est *E-inconsistante*. Le théorème suivant éclaire la relation entre la notion d'E-interprétation et celle de validité d'un ensemble de clauses.

Théorème: (voir, par exemple, Chang et Lee 1973): un ensemble de clauses S est E-inconsistant ssi $S \cup K$est inconsistant.

3.1. Relations de Réduction Définies par des E-interprétations

Comme les termes clos sont totalement ordonnés, un atome égalitaire clos $s=t$ peut s'orienter par exemple en $s \rightarrow t$ (lorsque $s>t$) et s'utiliser comme une règle de réduction.

Soit w et v deux atomes clos et I une E-interpretation partielle sur D. Nous dirons que w est *I-réductible en v par* $s=t$ (dénoté par $w \rightarrow_{Iv}$) s'il y a un atome $(s=t) \in D$ tel que:

$w=w[s]$, $s>t$, $w>(s=t)$, $I(s=t)=V$, et $v=w[t]$.

La clôture réflexive-transitive de $\rightarrow_I$ sera notée $\rightarrow_I *$. Un atome qui n'est pas *I-réductible* sera dit *I-irréductible*. La proposition suivante affirme que pour tester l'I-irréductibilité d'un atome, il suffit de considérer les égalités I-irréductibles:

Théorème de Réduction: w est I-réductible ssi il est I-réductible par une égalité qui est I-irréductible.

Preuve: supposer que w est I-réductible et considérer la plus petite égalité qui I-réduit w.

3.2. Arbres E-sémantiques Transfinis.

Comme dans (Peterson 1983), il est possible de construire par induction les E-interprétations.

Théorème: Soit $I : W_{\alpha+1} \rightarrow \{V,F\}$ tel que I est une E-interprétation sur W_α. Soit J la restriction de I à W_α. Alors I est une E-interprétation sur $W_{\alpha+1}$ ssi:

(1) A_α est J-réductible en C et $I(A_\alpha) = I(C)$, *ou*
(2) A_α est J-irréductible, de la forme $t=t$, et $I(A_\alpha) = V$, *ou*
(3) A_α est J-irréductible et n'est pas de la forme $t=t$.

Remarquons que le cas 3 entraine que $I(A_\alpha)$ peut prendre aussi bien la valeur V que la valeur F. Le résultat précédent peut s'interpréter comme une propriété de convergence; en effet le cas (1) affirme implicitement que tous les C obtenus en réduisant A_α possèdent la même valeur de vérité dans I.

Preuve: La nécessité de la condition est une simple conséquence de la définition des E-interprétations. Montrons la suffisance de la condition. Pour simplifier les notations, renommons B l'atome A_α. Supposons (1), (2), (3) vérifiés. Prouvons les résultats suivants:

P1: si B=(t=t) alors I(B)=V.
P2: si B=(s=t), I(B)=V, A[s]<B et A[t]<B alors I(A[s])=I(A[t]).
P3: si B=B[s], B[t]<B, s=t <B, et I(s=t) =V alors I(B)=I(B[t]).

Prouvons d'abord P3. Comme B[t]<B[s], nécessairement s>t. Mais B est J-réductible (car I(s=t)=V). Il existe donc C tel que $B \rightarrow_J C$ et I(B)=I(C). Il reste à démontrer que I(B[t])=I(C). La technique utilisée est celle des preuves de confluence de systèmes de réécriture (Knuth Bendix 1970)(Huet 1980). Supposons que B se J-réduit en C par l=r, avec l>r. Soit n et m les occurrences respectives de s et l dans B. Nous pouvons écrire: B = B[n←s, m←l].

Cas 1: aucune des deux occurrences n et m n'est préfixe de l'autre.

Alors C=B[n←s, m←r] et B[t]=B[n←t, m←l]. Par définition des E-interprétations, I(B[t])=I(C). Par conséquent, I(B)=I(B[t]).

Cas 2: n est préfixe de m.

Alors l est un sous-terme de s. Donc, B=B[s[l]] et C=B[s[r]]. Comme I(l=r)=V, nous avons I(s[l]=t)=I(s[r]=t)=I(s=t)=V. De même: I(B[s[r]])=I(B[t]). Donc I(B)=I(B[t]).

Cas 3: m est préfixe de n.

Ce cas se résout comme le cas 2.

Prouvons P1. Si B=(t=t) et B est J-réductible, alors B peut-être I-réduit en un atome I-irréductible r=r. D'après P3, I(t=t)=I(r=r)=V. Si B est J-irréductible, P1 est trivial par (2).

Prouvons P2. Supposons que s>t. Comme s est un sous-terme de A, d'après la propriété O5

des CSO, A[s]<B implique que A[s] est de la forme s=r avec r<t. Donc A[t] est de la forme t=r. Montrons que I(A[s])=V ssi I(A[t])=V. Si I(A[s])=I(s=r)=V, alors B est J-réductible et donc par P3, I(l=r)=I(B)=V. D'où I(A[s])=I(A[t])=V. D'autre part si I(A[t])=V, alors B est J-réductible en s=r par t=r et de nouveau, P3 implique I(s=r)=I(B)=V. Nous avons ainsi montré que si l'une des égalités I(A[s])=V ou I(A[t])=V est vérifiée, l'autre l'est également. Q.E.D.

Un arbre sémantique transfini est simplement l'ensemble de toutes les E-interprétations partielles ordonnées par la relation de prolongement des applications ◁. De manière plus formelle, soit I et J deux interprétations partielles, de domaines W_α et W_β, alors

$I \triangleleft J$ si $\alpha < \beta$ et $J|_{W_\alpha} = I$. Si $I \triangleleft J$, on dit que J est une *extension* de I. L'ordre ◁ a les propriétés suivantes:

(1) ◁ est bien fondé.
(2) Si I est une E-interprétation partielle sur W_α, alors I a un ou deux successeurs (selon la I-réductibilité de A_α).
(3) Si I est une E-interprétation partielle sur W_α et α admet un prédecesseur pour <, alors $I|_{W_{\alpha-1}}$ est le prédécesseur de I pour l'ordre ◁.

(1) est évident, car l'ensemble des segments initiaux de l'ensemble bien-fondé $A(P,F)$ est aussi bien-fondé pour la relation d'inclusion des ensembles. (2) est une conséquence simple de la construction inductive des E-interprétations; en effet toute E-interprétation sur W_α peut s'étendre d'au plus deux manières à $W_{\alpha+1}$.

Par convention, lorsqu'une E-interprétation partielle I sur W_α admet deux descendants, celui de gauche (resp.droite) attribue la valeur V (resp.F) à l'atome W_α. La Figure 1 représente un exemple d'arbre E-sémantique transfini.

Résumons les propriétés des arbres E-sémantiques transfinis:

Proposition :

- Pour un CSO sur A (P,F), l'arbre E-sémantique associé est unique.
- Si l'ordre est transfini, l'arbre est également transfini.
- L'arbre E-sémantique contient toutes les E-interprétations partielles, chacune étant représentée par un chemin de l'arbre ayant pour origine la racine.

Comme chaque E-interprétation partielle I représente une unique branche dans l'arbre sémantique transfini, on utilisera le même symbole I pour dénoter un noeud dans l'arbre, qui se trouve à l'extrémité de la branche représentée par l'interprétation I.

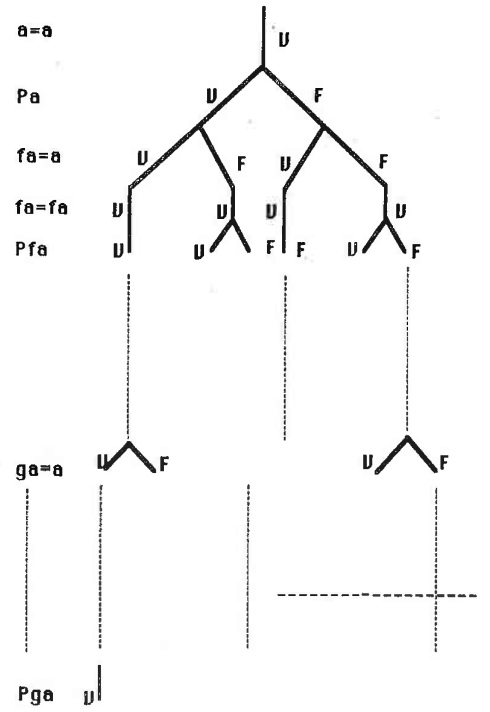


Figure 1

4. Une Méthode pour établir la Complétude Réfutationnelle.

Nous présentons maintenant une méthode de preuve utilisant les arbres sémantiques transfinis décrit plus haut.

Une stratégie de démonstration automatique P = <I,Σ> est formée d'un ensemble de règles d'inférence I, et d'un plan de recherche Σ (Kowalski 1970). Les règles d'inférences stipulent les conséquences qui sont immédiatement déductibles des données, et le plan de recherche spécifie sur quelles données et dans quel ordre sont appliquées les règles d'inférences.

Dans cette section, nous considérons seulement des règles qui construisent de nouvelles conséquences (et donc ne suppriment ni altèrent les données antérieures). Un ensemble de règles d'inférence est complet pour la réfutation (complet, pour abréger) si, quel que soit l'ensemble

inconsistant de clauses S, la stratégie de recherche en largeur d'abord admettant I comme ensemble de règles d'inférences engendre une contradiction. D'autres stratégies de recherche et des règles d'inférence qui peuvent altérer les données, comme la simplification et la subsomption seront considérées plus loin.

La recherche en largeur d'abord revient à considére la fermeture du processus d'inférence. Soit *INF* un ensemble de règles d'inférence et S un ensemble de clauses. *INF*(S) dénote l'ensemble des clauses obtenu en ajoutant à S toutes celles que l'on peut engendrer en appliquant *INF* à S.

Soit $INF^{n+1}(S) = INF(INF^n(S))$, et $INF^*(S) = \bigcup_n INF^n(S)$. Quand il n'y a pas d'ambigüité sur *INF*, on écrit S^* au lieu de $INF^*(S)$.

Proposition: *INF* est complet pour la réfutation si pour tout ensemble inconsistant de clauses, *NIL* (la clause vide) appartient à S^*.

4.1. Arbres Sémantiques Consistants.

Soit un arbre E-sémantique *ET*, l'*arbre sémantique consistant de S*, noté $MCT(S)$, est le sous-arbre maximal de *ET* tel que :

pour tout noeud I de $MCT(S)$, toute clause C de S, et toute substitution close θ, si les atomes de $C\theta$ appartiennent au domaine de I, alors $I(C\theta)=V$.

Si I falsifie un $C\theta$ mais aucun de ces ancêtres (pour l'ordre $\triangleleft$) ne falsifie une instance d'une clause de S, alors I est un *noeud d'échec*. En particulier, si J est le dernier noeud d'une branche de l'arbre sémantique consistant, alors toute extension de J est un noeud d'échec.

Une *suite croissante de noeuds* d'un arbre E-sémantique est un ensemble de noeuds $\{I_i\}_{i<\alpha}$ où $\alpha \leq \lambda$, λ étant l'ordinalité de la base de Herbrand, tel que chaque I_i est défini sur W_i, et I_j est une extension de I_i si $i<j<\alpha$. La *limite* d'une suite croissante de noeuds est la E-interprétation partielle $I_{sup\alpha}$, extension des I_i, pour $i<\alpha$. Une propriété importante des MCT's est le

Lemme de Clôture: la limite d'une suite croissante de noeuds de $MCT(S)$ appartient à $MCT(S)$.

Corollaire 1: si l'interprétation partielle I, de domaine W_α, est un noeud d'échec, alors α est soit 0 soit un ordinal successeur.

Corollaire 2: soit S un ensemble de clauses, alors S est E-inconsistant ssi toute branche de $MCT(S)$, considérée comme une interprétation partielle, admet une extension stricte.

Les deux corollaires sont des conséquences faciles du lemme de clôture. Le corollaire 2 signifie que toute extension d'une branche maximale de $MCT(S)$ doit être un noeud d'échec. Le Corollaire 1 signifie qu'un noeud d'échec ne peut apparaitre à la limite. Par conséquent, quand un noeud d'échec figure dans l'arbre, il admet toujours un prédécesseur immédiat. Ce fait sera utilisé dans nos preuves. Le Corollaire 2 joue dans notre méthode un rôle analogue au Théorème de Herbrand pour les autres méthodes à base d'arbres sémantiques. Remarquons encore que, d'après ce corollaire, s'il existe une branche maximale I de $MCT(S)$ qui n'admet pas d'extension, alors S doit être

consistant, car ce I ne falsifie aucune clause de S et se trouve défini sur toute la base de Herbrand.

4.1.1. Une Remarque sur les Arbres Transfinis.

Dans la méthode classique des arbres sémantiques (Kowalski Hayes 1970) (Peterson 1983), les auteurs introduisent habituellement une notion d'arbre sémantique clos, qui est essentiellement le plus petit sous-arbre de l'arbre sémantique permettant de montrer l'inconsistance d'un ensemble donné de clauses. Si, nous ajoutons à un arbre sémantique consistant l'ensemble des noeuds d'échec nous obtenons la notion d'arbre sémantique fermé, sur laquelle s'appuient les autres méthodes.

A la base de la plupart des méthodes de preuve par réfutation se trouve le théorème de Herbrand, qui assure qu'un ensemble de clauses est inconsistant ssi il existe un ensemble *fini* d'instances de ces clauses qui est inconsistant au sens du calcul des propositions. Le théorème de Herbrand permet donc de traiter un domaine infini par des méthodes finies. Pour prouver la complétude d'une stratégie de démonstration automatique, on commence, en général, par supposer l'existence d'un ensemble fini d'instances, comme ci-dessus, puis on applique les règles d'inférence pour faire décroitre l'ensemble suivant une mesure bien fondée. Comme le processus ne peut se poursuivre indéfiniment, la stratégie envisagée est donc complète. Dans les méthodes classiques d'arbres sémantiques, les inférences se traduisent par un rétrécissement de l'arbre fermé construit à partir de l'ensemble fini d'instances. La taille de l'arbre sert de mesure bien fondée. S'il est toujours possible de rétrécir l'arbre, alors, par induction, la stratégie est complète.

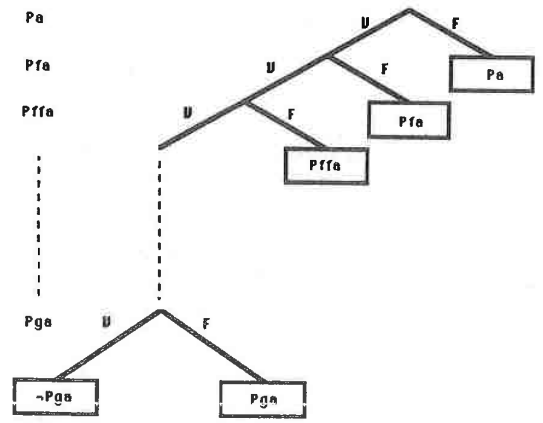


Figure 2

Notre méthode interdit l'usage du théorème de Herbrand. En effet, les arbres sémantiques consistants peuvent être infinis. Il en va de même pour l'ensemble des instances de clauses qui leur correspond. Considérons, par exemple, l'ensemble de clauses $S = \{\neg P(g(x), P(y)\}$. Supposons que le langage admet deux symboles de fonctions unaires f et g, une constante a, et que les objets sont comparés avec le RPO construit sur la précédence $a<f<g$. Alors la base de Herbrand s'ordonne comme suit:

$\{P(a),P(fa),P(ffa), ...,P(ga), P(gfa), P(gffa), ..., P(gga), ... \}$.

La Figure 2 représente l'arbre sémantique consistant associé, avec l'ensemble infini de clauses $\{P(a),P(fa), ..., P(ga), \neg P(ga)\}$ réfutant les noeuds d'échec. D'autre part, les deux instances closes $\{P(ga), \neg P(ga)\}$ suffisent à prouver l'inconsistance de S, mais l'arbre fermé associé n'est plus minimal.

Notre méthode n'utilise que le corollaire 2, qui est plus faible que le théorème de Herbrand. Remarquons enfin qu'en l'absence du prédicat d'égalité, même lorsque la base de Herbrand est ordonnée de manière transfinie, il est inutile de construire des arbres sémantiques infinis.

4.1.2. Le Plan de Preuve.

Rappelons que S^* représente $INF^*(S)$. L'étude précédente conduit à la définition suivante de la complétude par les arbres E-sémantiques :

Théorème fondamental: *INF* est complète ssi $MCT(S^*)$ est vide quand S est E-inconsistant.

Le théorème suggère la méthode suivante de preuve par contradiction: supposons que *INF* n'est pas complète; alors il existe un ensemble E-inconsistant de clauses S tel que $MCT(S^*)$ n'est pas vide. Nous définissons par induction (transfinie), une suite croissante particulière de noeuds de $MCT(S^*)$, de limite I. (Cette construction varie suivant *INF*). Comme S est E-inconsistant, toute extension de I falsifie une clause C de S^*. Nous appliquons une règle de *INF* pour déduire une autre clause D qui est falsifiée I. Mais comme S^* est l'ensemble de toute les conséquences que l'on peut déduire de S (par *INF*), I, appartenant à $MCT(S^*)$, ne peut pas falsifier D. Nous obtenons une contradiction, à moins que $MCT(S^*)$ ne soit vide.

Ce raisonnement peut aussi bien s'appliquer aux autres méthodes d'arbres sémantiques (Kowalski Hayes 1970). Inversement, le raisonnement par récurrence applicable aux arbres sémantiques conventionnels ne peut s'appliquer aux arbres transfinis, car ils sont transfinis non seulement en longueur, mais aussi en largeur. Dans la suite, nous appliquons cette technique à différentes stratégies.

CHAPITRE 3

Stratégies de Paramodulation.

La méthode de preuve du chapitre précédent sera appliquée dans ce chapitre à deux stratégies de paramodulation. Avant de les décrire, notons leurs points communs qui sont fondamentaux pour l'efficacité:

(a) les axiomes de réflexivité fonctionnelle ne sont pas requis.

(b) la paramodulation est utilisée sous une forme raffinée : un terme ne peut être remplacé par un terme plus complexe au cours d'une étape de paramodulation; en particulier, on ne paramodule jamais dans une variable.

(c) les stratégies restent complètes en présence de règles d'inférence réductrices comme la démodulation ou la subsomption. Ce dernier point sera traité au chapitre 4.

La première des stratégies que nous appelons *paramodulation ordonnée* exploite l'ordre sur l'univers de Herbrand, pour limiter le nombre d'inférences possibles. Les ordres de simplification sont utilisés pour comparer les littéraux à l'intérieur d'une clause et les membres d'une équation. La stratégie de paramodulation ordonnée peut se décrire de la manière suivante: une résolution ou une paramodulation doit toujours concerner les littéraux maximaux des clauses parentes. De nombreux auteurs ont obtenu des raffinements de la résolution en attribuant des priorités aux littéraux dans chaque clause: ces priorités sont définies à l'aide d'un ordre stable par instanciation (*A-ordering*) sur l'univers de Herbrand dans (Slagle 1967)(Kowalski Hayes 1970)(Joyner 1976)(Reiter 1971); des poids arbitraires sont affectés aux littéraux dans la méthode de *locking* de Boyer (1971). La méthode de locking a été étendue à la paramodulation par (Lankford 1972) et (Fribourg 1983) .

La deuxième stratégie, dite *stratégie positive*, exige que l'une des deux clauses impliquées par une résolution ou une paramodulation soit positive. C'est un cas particulier de *stratégie sémantique* ou *stratégie avec ensemble de support* (Wos Robinson 1968). La restriction d'une stratégie positive à un ensemble de clauses de Horn est une *stratégie unitaire* (Henschen Wos 1974). En corollaire nous obtenons la complétude de la stratégie unitaire pour les clauses de Horn. Ce résultat a été obtenu indépendamment par (Bachmair et al.1987) . Les auteurs, à la suite des travaux d'Etienne Paul (1986), utilisent une représentation équationnelle des clauses de Horn, à laquelle ils appliquent un algorithme de complétion sans échec.

1. Règles d'Inférence.

Introduisons maintenant les règles d'inférence de la *stratégie de clauses ordonnées*. Comme d'habitude, nous supposons que > est un ordre de simplification complet.

O-Factorisation:
Si $L_1,...,L_k$ sont des littéraux d'une clause C qui sont unifiables avec pour unificateur principal σ, et si pour tout atome A de $C-\{L_1, L_2,..., L_k\}$, $L_1\sigma \not\geq A\sigma$, alors $D = C\sigma - \{L_2\sigma,...,L_k\sigma\}$ est un *O-facteur* de C.

O-Résolution:
Soit $C_1 = L_1 \vee D_1$ et $C_2 = \neg L_2 \vee D_2$ deux clauses L_1 et L_2 unifiables, d'unificateur principal σ. Supposons de plus que, $L_1\sigma \not\geq A\sigma$ pour tout atome A de D_1 et $L_2\sigma \not\geq A\sigma$ pour tout atome A de D_2, alors $D = D_1\sigma \vee D_2\sigma$ est un *O-résolvant* de C_1 et C_2

O-Paramodulation:
Soit $C_1 = (s=t) \vee D_1$ et $C_2 = C_2[n \leftarrow r]$, où r est un sous-terme non-variable à l'occurrence n du littéral L de C_2. Si
(1) $s\sigma = r\sigma$ avec σ unificateur principal de s et r,
(2) $s\sigma \not\geq t\sigma$,
(3) $(s=t)\sigma \not\geq A\sigma$, $\forall A \in D_1$, et
(4) $L\sigma \not\geq A\sigma$, $\forall A \in C_2 - \{L\}$,
alors $C = C_2[n \leftarrow t]\sigma \vee D_1\sigma$ est un *O-paramodulant de C_1 dans C_2 à l'occurrence n.*

1.1. Remarque.

Comme un CSO est total sur les objets sans variables, on peut remplacer $\not\geq$ dans la définition des règles d'inférence ci-dessus par > quand les clauses n'ont pas de variables. Dans ce cas, toute clause déduite est plus petite qu'au moins l'un de ses parents. (Les clauses sont comparées par l'extension multi-ensemble de > sur leurs ensembles d'atomes.) D'après la bonne fondation de >, seul un nombre fini de clauses peut être engendré à partir d'un ensemble fini de clauses closes. En admettant que la stratégie est complète (nous le prouvons plus tard), on en déduit facilement qu'elle fournit aussi une procédure de décision pour les clauses closes.

1.2. Exemples

Si les prédicats sont ordonnés de la manière suivante:

$$P_n > P_{n-1} > ... > =$$

et si le CSO compare les symboles de prédicats d'abord, alors tout atome égalitaire est plus petit que n'importe quel atome non égalitaire. Donc il n'est jamais nécessaire d'appliquer une O-paramodulation avec une clause active contenant un atome non égalitaire. De même, une O-résolution s'applique seulement sur les littéraux ayant les plus grands symboles de prédicat dans leurs clauses respectives.

Considérons par exemple l'ensemble de clauses :

$Ra \vee b=a$	c1
$\neg Qa$	c2
$\neg Ra$	c3
$\neg Pa$	c4
$Pb \vee Qb$	c5

avec la précédence suivante sur les prédicats $P > Q > R >$ = et $b>a$ sur les fonctions, nous obtenons la réfutation suivante:

$b=a$	O-res. entre c1 et c3	c6
$Pb \vee Qb$	O-para. de c6 dans c5	c7
Qb	O-res. entre c7 et c4	c8
Qa	O-para. de c6 dans c8	c9
NIL	O-res. entre c9 et c2	c10

Une autre manière de parvenir à une contradiction est d'utiliser c6, $b=a$, comme une règle de réécriture$b \rightarrow a$, pour remplacer chaque occurrence de b par$a$ juste après avoir engendré c6. Ainsi, c5 sera remplacé par $Pa \vee Qa$ avant que c7 ne soit engendré. Ce type de règle qui supprime une clause ou la remplace par une clause plus simple, améliore considérablement l'efficacité de la procédure. Ces stratégies seront étudiées plus en détail au Chapitre 4.

Considérons maintenant les clauses :

$P(0)$	c1
$\neg P(x) \vee P(s(x))$	c2

Si la règle classique de résolution est appliquée, on produit une infinité de clauses: $P(s(0))$, $P(s(s(0)))$,..., $P(si(0))$,.... La stratégie des clauses ordonnées, par contre, ne produit aucun résolvant. Car $P(x)<P(s(x))$ quel que soit le CSO utilisé. Donc, la procédure s'arrêtera toujours en détectant la consistance.

Dans l'exemple suivant, on compare les arguments d'abord, et on suppose que 1>0.

$x=x$	c0
$f(x,1) \rightarrow x$	c1
$f(0, x) \rightarrow 0$	c2
$Q(x,y) \vee g(x, y) \rightarrow 0$	c3
$\neg Q(x, y) \vee g(x, y) \rightarrow 1$	c4
$\neg Q(x, y) \vee Q(x, 1)$	c5
$\neg Q(x,1) \vee Q(h(x), 1)$	c6
$f(g(a,b), g(h(a),1) \neq g(a,b)$	c7

Le symbole $\rightarrow$ est logiquement équivalent à =. Nous écrivons $\rightarrow$ au lieu de = quand un membre de l'équation est plus grand que l'autre pour l'ordre considéré. Notons que dans c3, c4 et c6, un seul des littéraux peut intervenir dans une O-résolution ou O-paramodulation.

Voici la preuve:

$g(x,y) \rightarrow 0 \vee g(x,y) \rightarrow 1$	O-res. entre c3 & c4	c8
$\neg Q(x,y) \vee g(h(x),1) \rightarrow 1$	O-res. entre c4 & c6	c9
$g(h(a),1) \rightarrow 0 \vee f(g(a,b),1) \neq g(a,b)$	O-par. de c8 dans c7	c10'
$g(h(a),1) \rightarrow 0$	O-res. entre c1 & c10'	c10
$f(g(a,b),0) \neq g(a,b)$	O-par.de c10 dans c7	c11
$g(x,y) \rightarrow 0 \vee g(h(x),1) \rightarrow 1$	O-res. entre c9 & c3	c12
$g(a,y) \rightarrow 0 \vee 1 \rightarrow 0$	O-par.de c10 dans c12	c13
$f(0,0) \neq 0 \vee 1 \rightarrow 0$	O-par.de c13 dans c11	c14'
$1 \rightarrow 0$	O-res. entre c1 & c14'	c14
$f(x,0) \rightarrow x$	O-par. c14 dans c1	c15
$g(a,b) \neq g(a,b)$	O-par. c15 dans c11	c16
NIL	O-res. entre c16 and c0	

1.3. Complétude des Stratégies de Clauses Ordonnées.

Nous prouvons que la O-résolution, O-factorisation, et O-paramodulation forment une stratégie complète pour le calcul des prédicats du premier ordre avec égalité. Appelons cet ensemble de règles *ORP*. Rappelons que pour un ensemble de clauses S, S^* représente sa clôture par l'application des règles , c'est-à-dire $ORP^*(S)$.

Théorème: Soit S un ensemble E-inconsistant de clauses contenant la clause $x=x$. Alors S^* contient la clause vide *NIL*.

Signalons l'absence des axiomes réflexifs fonctionnels. De plus il est inutile de paramoduler dans une variable.

Preuve :

Supposons que *ORP* n'est pas complète, alors par le Théorème Fondamental il existe un ensemble E-inconsistant S tel que $MCT(S^*)$ est non vide. Suivant le plan de démonstration établi dans une section précédente, nous définirons par induction, une suite de noeuds de $MCT(S^*)$. (Pour la stratégie considérée, la branche droite). Nous montrerons ensuite que cette suite est vide, ce qui achèvera la preuve.

Soit A_0 le plus petit atome clos. Nous construisons la suite de E-interprétations partielles de la manière suivante. Soit d'abord $I_0 = \emptyset$ (l'interprétation vide). Supposons que I_i a été définie pour tout A_i avec $i < \alpha$ (ce qui implique qu'ils sont tous dans $MCT(S^*)$). Définissons I_α , si possible, par:

Supposons que α n'est pas un ordinal limite. Autrement dit, α admet un prédécesseur α -1. Soit K l'interprétation $I_{\alpha-1}$. Plusieurs cas se présentent:

(1) Si K n'a aucun successeur dans $MCT(S^*)$, alors la suite est achevée.

(2) Si K a un successeur J dans $MCT(S^*)$, I_α est égal à J.

(3) Si K a deux successeurs L et R dans ET, avec $L(A_{\alpha-1})$=V et $R(A_{\alpha-1})$= F, alors :

(3.1.) si R est un noeud d'échec, alors $I_\alpha = L$
(3.2) si R n'est pas un noeud d'échec, alors $I_\alpha = R$.

Supposons que α est un ordinal limite. Nous définissons simplement I_α comme $\lim_{i\to\alpha} I_i$ ou, plus précisément $I_\alpha(A_i)=I_{i+1}(A_i)$ pour $i<\alpha$. Cette définition est licite car pour i plus petit que α, $i+1$ est aussi plus petit que α (car α est un ordinal limite). I_α est une E-interprétation partielle car $\{I_i\}_{i<\alpha}$ est une suite croissante de E-interprétations partielles. D'après le lemme de clôture, I_α appartient à $MCT(S^*)$.

La suite ainsi construite n'est pas vide car $MCT(S^*)$ n'est pas vide. Elle n'admet pas non plus d'élément défini sur toute la base de Herbrand. Sinon soit I un tel élément de $MCT(S^*)$. Comme S est E-inconsistant, alors il existe une clause close C dans S ayant une instance D telle que $I \not\models D$. Soit A_β le plus grand atome de D. Soit $I_{\beta+1} = I_{\beta+1}$. Comme I est une extension de $I_{\beta+1}$ et $I_{\beta+1}$ est défini sur tous les atomes $\leq\beta$, $I_{\beta+1} \not\models D$. Cela contredit l'hypothèse selon laquelle I (et donc $I_{\beta+1}$) est dans $MCT(S^*)$.

Soit $K=I_\alpha$ la E-interprétation définie ci-dessus. Donc, K est définie sur tous les A_i où $i<\alpha$, K est dans $MCT(S^*)$, et toute extension de K est un noeud d'échec. Notons l'atome A_α par B. D'après le lemme de clôture, l'un des trois cas suivants doit avoir lieu (Figure 3):

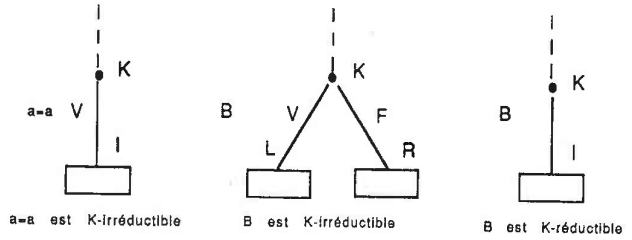
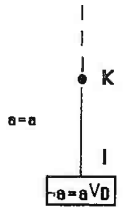


figure 3

Cas 1: K admet une extension I et B est K-irreducible.
(Ainsi, B est de la forme $a=a$, pour un certain a.)
Cas 2: K admet deux extensions L et R (donc, B est K-irréductible).
Cas 3: K admet deux extensions et B est K-réductible.

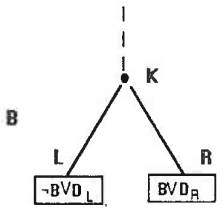
Nous devons prouver que dans chaque cas il existe une clause de S^* qui falsifie l'interprétation K, contredisant l'appartenance de K à $MCT(S^*)$. Nous présentons d'abord la preuve dans le cas clos. Autrement dit, toutes les clauses de S sont supposées sans variables.(En particulier S contient tous les atomes égalitaires $a=a$, obtenus par instanciations de $x=x$). La preuve sera ensuite étendue au cas général grâce à des lemmes de relèvement.

Cas 1



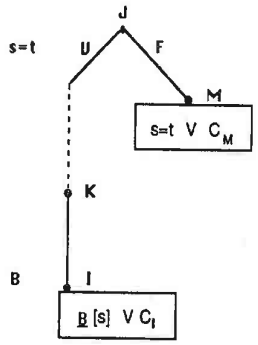
L'atome B est une égalité $a=a$ K-irréductible (et $I(a=a)=$ V, par définition), a. étant un certain terme clos. Comme I est un noeud d'échec pour S^*, il y a une clause $C \in S$ telle que $I(C) = \mathbf{F}$. Comme K ne falsifie pas C ($K \in MCT(S^*)$, $C = \neg(a=a)\vee D$ pour un certain D, dont les atomes sont plus petits que $a=a$. Il est clair que $K(D)= \mathbf{F}$ car K et I diffèrent seulement par leurs valeurs sur B. Puisque $a=a$ appartient à S^*, $a=a$ et $\neg(a=a)\vee D$ engendrent un O-resolvant D. D est donc dans S^* (S^* contient toutes les conséquences obtenues par itération des règles d'inférence). Donc K ne peut appartenir à $MCT(S^*)$ (car $K(D)=\mathbf{F}$), ce qui est contradictoire. Le Cas 1 n'a donc pas lieu.

Cas 2



Le noeud K a deux extensions qui sont des noeuds d'échec pour S^*. Ainsi, $L(C_L)=R(C_R)= \mathbf{F}$ pour des clauses C_L, C_R de S. Comme K ne falsifie pas C_L ou C_R, on a $C_L = \neg B \vee D_L$ et $C_R = B\vee D_R$. De plus, tous les atomes de D_L et D_R sont plus petits que B, et $K(D_L)=K(D_R)=K(D_L\vee D_R)= \mathbf{F}$. Par O-résolution de C_L et C_R, $D_L \vee D_R$ est un O-résolvant de C_L et C_R. Donc $D_L\vee D_R$ appartient à S^*, et K ne peut appartenir à $MCT(S^*)$. Le Cas 2 n'a pas lieu.

Cas 3



L'atome B est K-réductible. Soit $s=t$ la plus petite équation telle que $s>t$, $B=B[s]$, et $K(s=t)=V$. On vérifie que l'atome $s=t$ is K-irréductible. Soit β l'indice de l'atome $s=t$ et J la restriction de K à W_β (définie sur les atomes plus petits que $s=t$). Soit M le successeur droit de J. D'après la construction de K (branche droite de $MCT(S^*)$), M est un noeud d'échec. On peut établir les faits suivants:

(1) M falsifie une clause $s=t \vee C_M$ de S^*, et chaque atome de C_M est plus petit que $s=t$.
(2) I falsifie une clause $\underline{B}[s] \vee C_I \in S^*$, où $\underline{B}[s]$ représente $B[s]$ si $I(B[s])= F$, et $\neg B[s]$ si $I(B[s]) = V$. De plus tout atome de C_I est plus petit que $B[s]$.
(3) $J(C_M) = K(C_M) = K(C_I) = F$.
(4) $I(\underline{B}[s]) = I(\underline{B}[t]) = K(\underline{B}[t]) = F$.

(2) et (3) sont obtenus comme dans les cas précédents. Expliquons (4). $I(s=t)=V$ implique que la valeur de vérité de $B[s]$ pour I est égale à celle de $B[t]$ pour I. Comme $B[t]<B[s]$, $B[t]$ doit appartenir au domaine de K. Par compatibilité de K et I, on obtient les équations de (4).

Par O-paramodulation, comme $s>t$, $s=t \vee C_M$ et $\underline{B}[s] \vee C_I$ engendrent $\underline{B}[t] \vee C_M \vee C_I$ qui appartient à S^*. Par (3) et (4) ci-dessus, $K(\underline{B}[t] \vee C_M \vee C_I) = F$, en contradiction avec l'hypothèse selon laquelle K n'est pas un noeud d'échec. Donc le Cas 3 est aussi impossible.

L'impossibilité des trois cas entraine une contradiction avec: $MCT(S^*) \neq \emptyset$. Donc $MCT(S^*)$ est vide, et par conséquent, ORP est complète.

2. Les Arguments de Relèvements.

Si les déductions effectuées sur les termes clos sont des *instances* de déductions effectuées sur les termes avec variables, alors la complétude sur les termes clos pourra immédiatement s'étendre au cas général avec variables. Les lemmes de relèvement montrent justement que chaque inférence

sur des instances closes de clauses reflète une inférence possible sur les clauses elle-mêmes.

2.1. Substitutions Irréductibles.

Alors que le relèvement de la résolution est facile, celui de la paramodulation est plus complexe. Illustrons le par un exemple. Soit $P(x,x,c)$ une clause et $c=a$ (avec c>a) une équation. Considérons l' instance $P(c,c,c)$ de $P(x,x,c)$. La paramodulation de $c=a$ sur le troisième argument de $P(c,c,c)$ engendre $P(c,c,a)$. Une paramodulation analogue de c=a sur la clause elle même $P(x,x,c)$, produit $P(x,x,a)$ qui admet $P(c,c,a)$ comme instance. Par contre si on effectue une paramodulation sur le premier argument de $P(c,c,c)$ on engendre $P(a,c,c)$. Comme le règle de paramodulation interdit de remplacer une variable, aucune inférence sur les deux clauses avec variable ne peut produire une clause ayant $P(a,c,c)$ comme instance. Cela motive la définition suivante.

Définition : Soit I une E-interprétation (partielle) et σ et θ deux substitutions. On dit que σ *est I-réductible en* θ, et on le note $\sigma \rightarrow_I \theta$, si σ est identique à θ sauf pour une variable x, et $I(\sigma(x)=\theta(x)) = V$, et $\sigma(x)>\theta(x)$. Si σ n'est I-réductible en aucune substitution, alors σ est *I-irréductible.*

Théorème de la Substitution Irréductible: Supposons que I est une E-interprétation (partielle), σ une substitution close, et C une clause telle que les atomes de $C\sigma$ appartiennent au domaine de I. Si $\sigma \rightarrow_I \theta$, alors $I(C\sigma) = I(C\theta)$

Preuve:

Soit θ une substitution telle que $\sigma \rightarrow_I \theta$. Par définition de la I-réduction, tous les termes de θ sont plus petits que les termes de σ. Donc tous les atomes de $C\theta$ appartiennent au domaine de I. Par définition d'une E-interprétation, un terme s de $C\sigma$ peut être remplacé par (I-réduit en) un terme t de θ seulement si I(s=t)=V, et la valeur de vérité de $C\sigma$ ne change pas au cours de ce remplacement. Donc la valeur de vérité de $C\theta$ reste égale à celle de $C\sigma$. Q.E.D.

Corollaire: Soit I une E-interprétation (partielle) et $C\sigma$ une clause dont les atomes sont dans le domaine de I. Alors il existe une substitution I-irréductible θ telle que $I(C\sigma)=I(C\theta)$.

Preuve:

Comme $\rightarrow_I$ est une relation noethérienne sur les termes, elle est aussi noethérienne sur les substitutions. On peut donc obtenir θ en normalisant σ par la relation de réduction $\rightarrow_I$. Q.E.D.

Le corollaire montre que l'on a seulement besoin de considérer les substitutions irréductibles. Cette propriété permet de relever la paramodulation, car elle permet de considérer dans les clauses closes uniquement les occurrences qui existaient au niveau général. Avant de détailler ce point, reprenons l'exemple avec $P(x,x,c)$ et $c=a$. Soit I une E-interprétation dont le domaine contient $P(c,c,c)$ et $c=a$. Comme $c>a$, le domaine de I contient aussi $P(a,a,c)$. Donc, si on utilise l'instance $P(a,a,c)$ plutôt que $P(c,c,c)$, toute paramodulation de $c=a$ dans $P(a,a,c)$ est instance d'une paramodulation de $c=a$ dans $P(x,x,c)$.

2.2. Les Lemmes de Relèvement.

Nous énonçons le lemme de relèvement pour la résolution.

Lemme de Relèvement de la Résolution (J.A. Robinson 1965): Soit C_1 et C_2 deux clauses sans variables communes, et $D_1 = P \vee D_1'$ et $D_2 = \neg P \vee D_2'$ deux instances closes de C_1 et C_2 respectivement. Soit $D = D_1' \vee D_2'$ (un résolvant de D_1 et D_2), alors de C_1 et C_2 on peut déduire une clause C par résolution et factorisation telle que D est une instance de C.

Preuve

La preuve est standard. Supposons que $C_1\sigma = D_1$ et $C_2\sigma = D_2$. De plus, supposons que $C_1 = P_1 \vee \ldots \vee P_k \vee C_1'$ et $C_2 = L_1 \vee \ldots \vee L_n \vee C_2'$ avec $P_1\sigma = \ldots = P_k\sigma = P$ et $L_1\sigma = \ldots = L_n\sigma = \neg P$. Alors il existe un unificateur principal δ tel que $P_1\delta = \ldots = P_k\sigma = \neg L_1\delta = \ldots = \neg L_n\delta$. De plus, il existe une substitution ρ telle que $\delta\rho = \sigma$. Maintenant, par factorisation sur C_1 et C_2, nous obtenons $E_1 = P_1\delta \vee C_1'\delta$ et $E_2 = L_1\delta \vee C_2'\delta$. Comme $P_1\delta\rho = \neg L_1\delta\rho$, $P_1\delta$ et $\neg L_1\delta$ sont unifiables avec pour unificateur principal θ et donc, il existe une substitution η telle que $\theta\eta = \rho$. Par résolution entre E_1 et E_2 nous construisons le résolvant $C = C_1'\delta\theta \vee C_2'\delta\theta$, qui a D comme instance puisque $C\eta = D$. Q.E.D.

Le relèvement de la paramodulation est analogue. Cependant, une étape de paramodulation ne peut pas se relever si le terme clos que l'on remplace dans la clause instanciée a une occurrence qui n'existe pas dans la clause générale. Nous énonçons un lemme de relèvement de la paramodulation qui tient compte du raffinement de l'ordre imposé aux termes et aux littéraux Pour simplifier, nous ne considérons pas les éventuelles étapes de factorisation requises.

Lemme de Relèvement de la O-Paramodulation: Soit C_1 et C_2 deux clauses sans variables communes et n une position non-variable dans C_2. Soit D un O-paramodulant de $C_1\sigma$ dans $C_2\sigma$ en n, où σ est une substitution close, alors il existe un O-paramodulant C de C_1 dans C_2 en n tel que D est une instance de C.

Preuve:

Soit C_1 la clause $(s=t) \vee C_1'$, C_2 la clause $L \vee C_2'$, et θ une substitution close telle que $L\theta > C_2'\theta$, $s\theta > t\theta$, et $(s=t)\,\theta > C_1'\theta$. De plus, nous supposons que n est une occurrence du littéral L. Soit C l'O-paramodulant $C_2\theta[n \leftarrow t\theta] \vee C_1'\theta$. Comme n est une occurrence de C_2, $C_2\theta/n = (C_2/n)\theta$. Ainsi, $(C_2/n)\theta = s\theta$ car $s\theta = C_2\theta/n$. Donc C_1/n et s sont unifiables. Soit σ leur unificateur principal; alors il existe une substitution ψ telle que $\theta = \sigma\psi$. De plus, nous avons $s\sigma \not\leq t\sigma$ (sinon, $s\sigma \leq t\sigma$ impliquerait $s\sigma\psi \leq t\sigma\psi$, en contradiction avec l'hypothèse selon laquelle $s\theta > t\theta$). Par le même type d'argument, $L\sigma \not\leq C_2'\sigma$. Donc il existe un O-paramodulant C de C_1 à l' occurrence n de C_2 tel que $C = (C_2[n \leftarrow t] \vee C_1')\,\sigma$, et $C\psi = (C_2[n \leftarrow t] \vee C_1')\sigma\psi = C_2[n \leftarrow t] \vee C_1')\theta = C_2\theta[n \leftarrow t\theta] \vee C_1'\theta = D$. Q.E.D.

2.3. Relèvement des Stratégies de Clauses Ordonnéees.

Pour utiliser le lemmme de relèvement de la paramodulation, il faut que chaque O-paramodulation sur une instance close $C\theta$ d'une clause C, s'applique à une occurrence non-variable de C. Par le corollaire de la section précédente, cette propriété sera satisfaite si l'on considère des substitutions irréductibles. Pour être plus précis, soit C une clause, $C\theta$ une instance close, et I une E-interprétation partielle telle que $I(C\theta) = F$. On suppose de plus que θ est I-irréductible, et qu'il existe une équation $s=t$ telle que $I\,(s=t) = V$ (et $s > t$). Supposons que s est un sous-terme de $C\theta$ à l' occurrence n, c'est-à-dire, $C\theta$ est I-réductible par $s=t$. Comme θ est I-irréductible, s ne peut pas être un sous-terme de l'image d'une variable de C par la substitution. Ainsi, l'occurrence n appartient déjà à C. En d'autres termes, chaque paramodulation dans $C\theta$ a lieu au-dessus de la partie substituée de $C\theta$. Donc nous pouvons appliquer le lemme de relèvement de la paramodulation.

Appliquons l'explication précédente au Cas 3 de la preuve de complétude. Supposons que $\underline{R}[s]$ ou C_I dans la preuve est une instance $C\theta$ d'une clause C de l'ensemble des clauses données. Supposons que θ est I-irréductible (sinon, par le Théorème de la Substitution Irréductible nous pouvons choisir une autre instance avec une substitution I-irréductible qui falsifie aussi I. Ainsi, le relèvement de la paramodulation est possible.

Le lemme de relèvement de la résolution ne peut pas s'appliquer directement à la O-résolution. Cependant, la propriété de stabilité O3 des ordres utilisés permet de l'adapter en conséquence.

2.4. Autres Stratégies basées sur un Ordre.

L'idée de raffiner la résolution en imposant un ordre sur les termes est très ancienne (Reynolds 1966, selon Loveland, 1978). A côté des méthodes simples comme l'ordre sur les symboles de prédicats, il y a des schémas plus complexes (Reiter 1971) et (Slagle Norton 1971) (clauses ordonnées), (Kowalski Hayes 1969) et (Luckham 1968) (A-orderings), et (Boyer 1971) (locking).

2.5. Inférences Bloquées.

Le blocage est un raffinement des règles d'inférence d'abord imaginé par (Slagle 1974) puis étendu par Lankford (Lankford 1975). Il sert également à obtenir des critères de paires critiques en théorie de la réécriture (Kapur et al. 1985) (Bachmair Dershowitz 1986). Intuitivement, le blocage n' autorise une inférence que lorsqu'(une partie d') une clause parente n'est pas réductible par une équation du système. De manière plus précise, un atome A est réductible par $l=r$ si A contient un sous-terme s tel que $s = l\sigma$, $l\sigma > r\sigma$, et $A > (l=r)\sigma$, où $>$ est le CSO utilisé par le système.

O-Résolution Bloquée

Soit $C_1 = L_1 \vee D_1$ et $C_2 = \neg L_2 \vee D_2$ deux clauses admettant un O-résolvant D. Alors D est un O-résolvant bloqué si $L_1\sigma$ n'est réductible par aucune clause unitaire $l=r$ de S (S étant l'ensemble des clauses du système au moment de l'inférence).

O-Paramodulation Bloquée

Soit $C_1 = (s=t) \vee D_1$ et $C_2 = C_2[n \leftarrow r]$, où r est un sous-terme non-variable à l'occurrence n du littéral L ($\in C_2$). Supposons de plus que $C = C_2[n \leftarrow t]\sigma \vee D_1\sigma$ est un O-paramodulant de C_1 dans C_2 en n. Alors C est un *O-paramodulant bloqué* si $(s=t)\sigma$ n'est réductible par aucune clause unitaire $l=r$ de S.

Le théorème suivant montre que les inférences bloquées sont suffisantes pour avoir la complétude.

Théorème: La O-factorisation, la O-résolution bloquée, et la O-paramodulation bloquée constituent une stratégie complète pour la réfutation.

Preuve:

La preuve est en fait déjà dans la preuve de complétude de la stratégie des clauses ordonnées. Etudions d'abord le cas 2 de cette démonstration. Comme le noeud K admet deux extensions, l'atome B doit être K-irréductible. Si il existe une équation (close) l=r dans S* telle que B est réductible par l=r, alors K(l=r)=**F**, sinon B serait K-réductible. Cependant, cele implique que S* aurait un noeud d'échec au-dessus de K, en contradiction avec l'hypothèse selon laquelle K appartient à MCT(S*).

Le cas 3 est analogue . L'atome (s=t) est M-irréductible. S'il existe l=r dans dans S* qui réduit s=t, alors de nouveau, M admet un prédecesseur qui est un noeud d'échec, en contradiction avec son appartenance à MCT(S*).

2.6. Une Restriction supplémentaire de la Factorisation

Une analyse détaillée de la preuve de complétude montre que les étapes de factorisation sont nécessaires seulement lorsqu'elles précèdent des étapes de résolution ou de paramodulation (Kowalski Hayes 1970).

3. Stratégies Positives.

Les stratégies positives imposent qu'une des clauses parentes de chaque résolution ou paramodulation soit positive. Elles sont des cas particuliers de stratégies *sémantiques*, ou avec *ensemble de support* (Wos Robinson 1968). Quand on la restreint à des clauses de Horn clauses, une stratégie positive se réduit à une stratégie unitaire, car toute clause de Horn positive est unitaire (Henschen Wos 1974).

Dans cette partie, nous montrons la complétude de la résolution positive et de la paramodulation positive sans les axiomes réflexifs fonctionnels. Nous pouvons même surimposer quelques restrictions dues à l'ordre utilisé. Ce résultat entraine la complétude de la stratégie unitaire positive pour les clauses de Horn. E.Paul (Paul 1985) a relié cette stratégie à la procédure de Knuth-Bendix et prouvé sa complétude sous réserve que chaque équation apparaissant dans la système soit orientable. Sa preuve n'est pas satisfaisante, car les équations non-orientables sont très fréquentes.

3.1. Règles d'Inférence.

Une clause est *positive* (*P-clause*) si elle n'a que des littéraux positifs. Comme d'habitude nous supposons que < est un CSO. La stratégie positive comporte les règles d'inférence suivantes:

O-Factorisation

Si $L_1,..., L_k$ sont des littéraux d'une clause C ayant un unificateur principal σ, et pour tout $A \in C - \{L_1, L_2,..., L_k\}$, $L_1\sigma \not\equiv A\sigma$. Alors $D = C\sigma - \{L_2\sigma,..., L_k\sigma\}$ est un *O-facteur* de C.

PO-Résolution

Soit $C_1 = \neg L_1 \vee D_1$ et $C_2 = L_2 \vee D_2$ deux clauses telles que
(1) C_2 est une P-clause,
(2) L_1 et L_2 ont un unificateur principal σ, et
(3) $\forall A \in D_2, L_2\sigma \not\equiv A\sigma$,
alors $D_1\sigma \vee D_2\sigma$ est un *PO-résolvant* de C_1 et C_2.

PO-Paramodulation

Soit $C_1 = (s=t) \vee D_1$ une P-clause et $C_2 = C_2[n \leftarrow r]$, où r est un sous-terme non-variable à l'occurrence n de C_2. Si
(1) $s\sigma = r\sigma$ avec σ unificateur principal de s et r,
(2) $s\sigma \not\equiv t\sigma$, et
(3) $\forall A \in D_1, (s=t)\sigma \not\equiv A\sigma$
alors $C = C_2[n \leftarrow t]\sigma \vee D_1\sigma$ est un *PO-paramodulant* de C_1 dans C_2 en n.

Remarquons que les stratégies positives, à l'opposé des stratégies de clauses ordonnées, n'impose l'ordre que sur les littéraux de la clause parente positive.

3.2. Exemples.

Considérons l'ensemble de clauses c1,...,c4, où a, b, c, d sont des constantes, f est une fonction unaire, et Q est un prédicat unaire. Nous utilisons l'ordre RPO avec la précédence Q>"=",a>b>c>d :

$f(a) \neq f(b) \vee Q(c)$		c1
$c \rightarrow d \vee \neg Q(c)$		c2
$a \rightarrow b \vee Q(c)$		c3
$f(c) \neq f(d) \vee \neg Q(c)$		c4
$c \rightarrow d \vee a \rightarrow b$	PO-res. c2 & c3	c5
$c \rightarrow d \vee Q(c)$	PO-par. de c5 dans c1 et res. avec x=x	c6
$c \rightarrow d$	PO-res. de c6 & c1 et O-fact.	c7
$\neg Q(c)$	PO-par. de c7 dans c4 et res. avec x=x	c8
$a \rightarrow b$	PO-res. c8 & c3	c9
Q(c)	PO-par. de c9 dans c1 et res. avec x=x	c10
NIL	PO-res. c8 & c10	

3.3. Complétude.

Montrons que la PO-résolution, PO-paramodulation, et O-factorisation forment un système complet. Nous appelons cette stratégie *POS*. Rappelons que pour un ensemble de clauses S, S^* représente la fermeture $POS^*(S)$.

Théorème: Soit S un ensemble E-inconsistant de clauses contenant l'atome $x=x$. Alors S^* contient la clause *NIL*.

Preuve:

Supposons que S soit E-inconsistant et $MCT(S^*)$ non vide. Nous construisons la branche droite I de $MCT(S^*)$ comme dans la preuve de complétude de la stratégie des clauses ordonnées. Rappelons que pour chaque *noeud* K précédant I, si K admet un successeur droit R, alors R est un noeud d'échec. Supposons que nous puissions montrer que ces R falsifient toujours une clause positive *K-irréductible*, alors on peut reprendre mot pour mot la preuve de la stratégie des clauses ordonnées, avec pour seule différence que l'on utilise les clauses positives falsifiées par R. Il rest seulement à démontrer le lemme suivant:

Lemme: Soit S un ensemble E-inconsistant de clauses contenant l'atome $x=x$. Soit R un noeud d'échec qui est le successeur droit d'un noeud K de la branche droite de $MCT(S^*)$. Alors R falsifie une clause positive K-irréductible de S^*.

Preuve:

La preuve du lemme procède par induction sur le domaine de R.

Base de l'induction :

Si R est définie sur W_1 (W_0 est l'ensemble vide et n'a pas besoin d'être considéré), R est seulement définie sur A_0 avec pour valeur **F**. Donc la seule clause close qui puisse être falsifiée par R est A_0, et $R(A_0) = \mathbf{F}$.

Etape d'induction :

Comme nous l'avons vu, si R est un noeud d'échec de domaine W_α, alors α doit être un ordinal successeur. Nous pouvons donc supposer que le domaine de R est $W_{\beta+1}$ pour un certain β

Soit $C_R = A_\beta \vee D$ la clause minimale (pour l'ordre < sur les objets clos) parmi toutes les clauses de S^* falsifiée par R. Nous montrons d'abord que C_R est K-irréductible, puis que C_R est positive. En notant que A_β est à la fois K-irréductible et positive, il suffit de prouver ces deux propriétés pour D.

Si D est K-réductible. Soit $A_\gamma = (s=t)$ (où s>t) la plus petite équation qui K-réduit D. Soit N l'ancêtre de K dont le domaine est W_γ. Alors par minimalité de $s=t$, N a un fils droit M et M est un noeud d'échec (voir Figure 5).

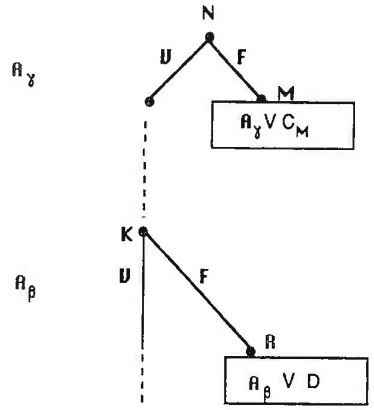


Figure 5

Par hypothèse d'induction, M falsifie une clause $(s=t)\vee C_M$ qui est N-irréductible, positive. Par PO-paramodulation de $(s=t) \vee C_M$ dans un sous-terme de D, nous engendrons une nouvelle clause $C_R' = A_\beta \vee D' \vee C_M$. C_R' appartient à S^*, est plus petite que C_R, est falsifiée *par R*. Cela contredit le choix de C_R. Ce qui montre que C_R est K-irréductible.

Nous montrons maintenant que $C_R = A_\beta \vee D$ est positive. Sinon, soit $\neg A_\gamma$ un littéral négatif de D (i.e. $D = \neg A_\gamma \vee D'$). Soit N l'ancêtre de K dont le domaine est W_γ. Par K-irréductibilité (et donc N-irréductibilité) de A_γ, N a un fils droit M qui est un noeud d'échec. Par hypothèse d'induction, M falsifie une clause positive N-irréductible $A_\gamma \vee C_M$. Par PO-résolution de $A_\gamma \vee C_M$ et C_R (qui contient $\neg A_\gamma$), nous obtenons la clause $A_\beta \vee C_M \vee D'$ qui est plus petite que C_R et est falsifiée par R. De nouveau, cela contredit le choix de C_R. Donc C_R est une clause positive.

3.4. Stratégie Positive Unitaire pour les Clauses de Horn.

Une*clause unitaire* est une clause n'ayant qu'un littéral. Une *clause de Horn* est une clause ayant au plus un littéral positif. Les règles d'inférences précédentes deviennent pour les clauses de Horn:

UP-résolution:

Soit P et $C = \neg L \vee D$ deux clauses, et $P\sigma = L\sigma$ avec σ unificateur principal, alors $D\sigma$ est un *UP-résolvant* de P et C.

UP-paramodulation:

Soit $(s=t)$ une clause unitaire et $C = C[n \leftarrow r]$ où r est un sous-terme non-variable de C_2. Si $s\sigma = r\sigma$ avec σ unificateur principal, et $s\sigma \$ t\sigma$, alors $D = C[n \leftarrow t]\sigma$ est un *UP-paramodulant* of $(s=t)$ dans C en n.

Ces deux règles forment une stratégie complète pour la réfutation. Remarquons que la règle de factorisation est inutile.

CHAPITRE 4

Complétude en présence de règles de réduction.

Nous appelons règles de réduction les règles d'inférences qui permettent de supprimer certaines clauses ou de les remplacer par des clauses plus simples. Ces règles sont essentielles pour améliorer les performances des procédures de démonstration automatique: leur rôle est de maintenir l'information sous une forme compacte (si possible, canonique), en écartant les redondances et les tautologies. Elles conduisent souvent à des preuves directes. Des règles de réduction, comme la subsomption ou la démodulation sont utilisés par de nombreux systèmes de preuve de théorèmes (ITP, SLOG,...) comme de puissantes heuristiques.

La règle de démodulation (appelée aussi simplification) consiste à remplacer dans une clause une instance du membre gauche d'une équation orientée par l'instance correspondante du membre droit. Nous généralisons cette définition pour pouvoir utiliser des équations non-orientables comme démodulateurs: l'orientation d'une équation est définie seulement après son instanciation, juste avant son éventuelle utilisation.

Bien que son intérêt ait été reconnu depuis longtemps (Wos et al. 1967)(Slagle 1974), les fondements théoriques de la démodulation se sont développés surtout à partir de la procédure de complétion de Knuth et Bendix (Lankford 1975) (Huet 1980,1981) (Bachmair et al. 1986). La notion de simplification est au coeur même des procédures de complétion: lorsqu'un système d'équations est canonique, cette seule règle d'inférence suffit pour résoudre le problème du mot.

Dans le cadre général de la logique clausale du premier ordre, très peu d'études envisagent la complétude des stratégies qui comportent des règles de réduction. L'aspect "non-monotone" de ces stratégies (certaines clauses peuvent disparaitre) est la source des difficultés rencontrées dans les preuves de complétude. Les ordres de simplification qui nous servent à orienter les équations, nous ont permis de traiter de manière rigoureuse la simplification, grâce notamment à leur propriété de bonne fondation.

La règle de subsomption consiste à supprimer toute clause qui contient une instance d'une autre clause. Elle s'apparente à la démodulation car elle repose également sur la notion de filtrage et restreint la taille de l'espace de recherche. Loveland (1978) et Kowalski (1970) l'ont étudiée du point de vue de la théorie de la preuve, qui est bien plus délicat que le point de vue sémantique. L'approche que nous proposons permet de traiter simultanément la démodulation et la simplification, ainsi que quelques règles de réduction voisines: simplification clausale, subsomption fonctionnelle

Dans ce chapitre, une règle d'inférence est une règle qui remplace un ensemble de clauses par un ensemble de clauses équivalent (dans la théorie de l'égalité).

1. Simplification et Subsomption.

Avec cette définition, nous pouvons considérer deux nouvelles règles d'inférence: la subsomption et la simplification.

1.1. Définitions.

Soit S un ensemble de clauses.

Subsomption: Si C_1 et C_2 sont deux clauses de S telles que C_1 a moins de littéraux que C_2 et $C_1\theta \subseteq C_2$ pour une substitution θ, alors on dit que C_2 est subsumée par C_1. La règle de subsomption s'applique en supprimant de S une clause qui est subsumée par une autre clause de S.

Certaines difficultés peuvent survenir du fait que la relation de subsomption n'est pas bien-fondée sur les clauses, même si elle est quotientée par la relation de renommage des variables: ainsi $P(f(x)) \vee P(f(z))$ et $P(u) \vee P(f(w))$ se subsument mutuellement sans être des variantes. Pour cette raison nous préférons utiliser une variante de la subsomption.

Stricte Subsomption: La clause C_1 subsume strictement la clause C_2 si C_1 subsume C_2 et C_2 ne subsume pas C_1. La règle de stricte subsomption s'applique en supprimant de S une clause qui est strictement subsumée par une autre clause de S.

Lemme (voir Loveland 1978): il n'existe pas de de suite infinie $C_0, C_1, \ldots$ telle que la clause d'indice i+1 subsume strictement la clause d'indice i, pour tout i.

Preuve

Supposons qu'il existe une suite comme dans le lemme. Chaque clause C_i subsume C_0. Cependant le nombre de clauses qui subsument une clause C_0 est fini, au renommage près des variables: en effet, le nombre de symboles d'une clause subsumant C_0 est borné par le nombre de littéraux de C_0 que multiplie le nombre maximal de symboles dans un littéral de C_0. La suite contient donc deux clauses variantes, disons C_n et C_{n+k} (k>0). Or, par transitivité de la relation de stricte subsomption, C_{n+k} subsume strictement Cn. Nous avons ainsi une contradiction. Q.E.D.

Pour la règle suivante, < représente un CSO.

Simplification: Si une équation s=t appartient à S et $C_2[s\theta]$ est une clause de S qui contient une instance $s\theta$ de s et $s\theta > t\theta$ et il existe un atome A de $C_2[s\theta]$ tel que $A > (s\theta = t\theta)$, alors la clause $C_2[t\theta]$ est une *simplification* de C_2 par s=t. La règle de simplification consiste à remplacer une clause par sa simplification.

Le format restrictif de la règle de simplification est nécessaire pour appliquer notre technique de

preuve de complétude. La restriction sur l'atome A n'est probablement pas utile comme le fait remarquer (Peterson 1983). Une restriction analogue apparait étrangement aussi dans des travaux récents sur la complétion (Bachmair et al. 1986).

Nous pouvons remarquer que, notre définition permet d'utiliser des équations non orientables pour simplifier: en effet il arrive que des équations non-orientables le deviennent après instanciation. Par exemple : f(x,x,y) =f(x,y,y), bien que non-orientable, peut simplifier P(f(g(a),g(a),a) en P(g(a),a,a).

1.2. Preuve de Complétude.

Nous considérons l'ensemble *INF* formé par les règles d'inférence suivantes: {O-résolution, O-factorisation, O-paramodulation, simplification, stricte subsomption}. Mais la technique s'applique aux autres systèmes considérés dans cette thèse. Nous supposons que les inférences sont appliquées de manière équitable (définition rigoureuse plus loin). Soit $S_0,S_1,S_2,\ldots$ une suite d'ensembles de clauses obtenue par application équitable des règles de INF. Nous appellerons une telle suite une **dérivation issue de S_0**. Comme certaines règles sont des règles de réduction, cette suite n'est pas croissante. On ne peut plus supposer qu'elle admet une limite. Donc la méthode du chapitre 2 ne s'applique pas directement. En effet si deux clauses permettent d'en inférer une troisième, il se peut que ces clauses n'appartiennent jamais au même S_n, donc que l'inférence n'ait jamais lieu. Pour éviter ce problème, nous allons montrer qu'une clause impliquée dans une réfutation peut être choisie de manière à ce qu'elle ne soit jamais simplifiée ou subsumée.

Hypothèses d'équité.

Comme dans l'algorithme de Knuth et Bendix, nous avons besoin d'hypothèses d'équité assurant qu'aucune inférence significative n'est éternellement différée.

Une dérivation $S_0,S_1,\ldots$ est **équitable** si:

s'il existe j tel que $R \in \cap_{i\geq j} RP(S_i)$ *alors* R *est subsumé par* $C \in \cup_{i\geq 0} S_i$

où RP(T) représente l'ensemble total des résolvants, paramodulants et facteurs que l'on peut obtenir (en une étape d'inférence) à partir de l'ensemble T.

La condition d'équité est classique dans le contexte des sytèmes de réécriture.

Nous dirons qu'une clause C est **persistante** (dans une dérivation Σ) si il existe un entier k tel que C appartient à tous les éléments de rang supérieur à k dans la suite Σ.

Notre preuve de complétude repose sur la proposition suivante:

Proposition: tout nœud d'échec de $\cup_{i\geq 0} S_i$ falsifie une clause persistante.

Admettons pour l'instant cette proposition. On en déduit le

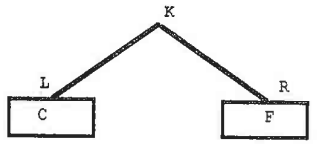
Théorème: soit S un ensemble E-inconsistant de clauses contenant $x=x$. Alors toute dérivation équitable issue de S contient la clause *NIL*.

Preuve du théorème:

Reprenons la preuve de complétude de la stratégie des clauses ordonnées. Soit une dérivation

équitable $S_0,S_1,S_2,\ldots$ issue de S un ensemble E-inconsistant de clauses contenant $x=x$.

Posons $S^*= \cup_{i\geq 0} S_i$. Supposons que MCT(S^*) ne soit pas vide. Soit K le dernier noeud de la branche droite de MCT(S^*). Supposons que K admet deux successeurs L et R dans ET, qui sont des noeuds d'échec pour S^*. Soit C une clause de S* falsifiée par L et F une clause de S* falsifiée par R.



Nous savons qu'il existe une clause RES dans RP({C,F}) falsifiée par K. Cette clause peut être obtenue par résolution de C et F. Grâce à la proposition ci-dessus, nous pouvons supposer que C et F sont persistantes, c'est-à-dire que :

$$C,F \in \cap_{i\geq j} S_i$$

On en déduit que

$$RES \in \cap_{i\geq j} RP(S_i)$$

L'hypothèse d'équité nous assure que RES est subsumée par une clause RES' de S*. Donc K falsifie la clause RES' de S* en contradiction avec le fait que K appartient à MCT(S*). Lorsque K admet seulement un seul successeur, la preuve est analogue.

Preuve de la proposition:

Soit *GR* l'application qui associe à tout sous-ensemble de clauses de $S^* = \cup_{i\geq 0} S_i$ l'ensemble de ses instances closes. Soit I un noeud d'échec de S* et S l'ensemble des clauses qui étiquettent I, c'est-à-dire:

$$\Sigma = \{G;\ G \in S^* \text{ et il existe } G1 \in GR(G) \text{ tel que } I(G1) = F\}$$

L'ensemble des élément minimaux de GR(S) pour l'extension multi-ensemble << de $<$, qui sont falsifiés par I est noté TG.

L'ensemble des clauses de Σ qui ont une instance dans TG est donc $GR^{-1}(TG)$; nous le noterons Π. Le sous-ensemble des clauses de Π qui sont minimales pour l'ordre de stricte subsomption sera noté Π' (i.e. l'ensemble de toutes les clauses de Π qui ne sont pas strictement subsumées par une autre clause de Π). Remarquons que $GR(\Pi) = GR(\Pi')=TG$.

Nous aurons besoin du lemme suivant:

Lemme: Si C appartient à Π' alors C est persistante.

Preuve du lemme:

Prouvons d'abord que C, un élément quelconque de Π' , n'est jamais simplifié. Sinon, il existe j tel que $C \in S_j$, $s \to t \in S_j$, C contient un sous-terme instance de s, ce que l'on note $C = C[s\sigma]$, pour une substitution σ et

$$S_{j+1} = (S_j - \{C\}) \cup \{C[t\sigma]\}$$

Comme C appartient à Π, il existe une substitution close θ telle que

$$I(C\theta)=F \text{ et } C\theta \text{ est une clause minimale dans } GR(\Sigma).$$

Par définition de la règle de simplification, $(s=t)\sigma < C$; donc, par stabilité de <, nous avons également $(s=t)\sigma\tau < C\theta$. Cependant nous ne pouvons pas avoir $I((s=t)\sigma\tau = F$: en effet, I est un noeud d'échec de S*, et, donc aucun prédécesseur de I dans ET ne peut falsifier une clause de S*.

Donc

$$I((s=t)\sigma\tau)=T$$

Comme I est une E-interprétation, nous en déduisons:

$$I(C[t\sigma]\theta) = I(C[s\sigma]\theta) = F$$

Mais

$$C[t\sigma]\theta < C[s\sigma]\theta$$

Nous obtenons donc une contradiction avec l'hypothèse selon laquelle $C\theta$ est minimal dans $GR(\Sigma)$.

Montrons maintenant que C n'est jamais subsumée. Supposons que C et CC appartiennent à S_j, que CC subsume strictement C et que $S_{j+1} = S_j - \{C\}$. Par définition, C ne subsume pas CC. Nous pouvons remarquer que CC appartient à Σ. Comme toute instance close de C contient une instance close de CC et C appartient à Π, CC appartient également à Π. Cependant C est dans Π' et donc ne peut être strictement subsumée par un autre membre de Π. Ceci achève la preuve du lemme.

1.3. Remarques.

Il est possible d'utiliser la règle de subsomption sans danger pour la complétude, en l'appliquant systématiquement à toute nouvelle clause apparaissant dans le système: chaque fois qu'une clause est sur le point d'être ajoutée aux données, on teste si une clause antérieure la subsume. Il est possible de modifier chacune des règles d'inférence pour que toute clause engendrée soit normalisée par les démodulateurs disponibles et non redondantes avec d'autres clauses. Soit X une règle d'inférence quelconque de type factorisation, résolution ou paramodulation. Il est toujours possible de remplacer cette règle X sans dommage pour la complétude par la :

NL_X_Règle: Soit C une clause obtenue par la règle X. Soit C' une clause obtenue en itérant la règle de simplification sur C jusqu'à ce qu'elle ne soit plus applicable. La règle NL_X permet d'ajouter C' (au lieu de C) à l'ensemble des clauses S s'il n'existe pas d'autre clause de S qui subsume C'.

2. Autres règles de réduction

Elimination des Tautologies: Une tautologie est une clause qui contient un littéral et sa négation. La règle d'élimination des tautologies autorise la suppression des tautologies.

En ajoutant cette règle, on ne détruit la complétude d'aucune des stratégies étudiées. En effet une tautologie n'est jamais falsifiée par une interprétation. Elle n'est donc jamais nécessaire pour construire une réfutation.

Simplification Clausale: Si un litteral négatif $\neg L$ appartient à S, alors on peut supprimer dans toute clause de S, une instance de L.

La règle de simplification clausale peut être simulée par une étape de résolution suivie par une étape de subsomption. Cependant lorsqu'on utilise un format raffiné de résolution, certaines simplifications clausales peuvent devenir impossibles à simuler de cette manière. Il est donc avantageux d'ajouter la règle de simplication clausale explicitement dans le système d'inférence. La complétude d'une stratégie est préservée lorsqu'on ajoute cette règle de réduction si l'application des règles non réductrices est équitable au sens vu dans la section précédente.

Subsomption Fonctionnelle:
Si une clause C de l'ensemble S contient le littéral $g[s]=g[t]$, et l'équation $l=r$ de S vérifie $l\sigma=s$ et $r\sigma=t$, alors on peut supprimer C de S.

La complétude est encore préservée, lorsqu'on ajoute cette règle. La démonstration est identique à celle utilisée dans la section précédente: il suffit de prouver que si C appartient à Π', elle n'est jamais subsumée fonctionnellement. Sinon, il existe un j tel que $C \in S_j$, $l=r \in S_j$, C contient un littéral de la forme $g[l\sigma]=g[r\sigma]$, pour une substitution σ et

$$S_{j+1} = S_j - \{C\}.$$

Comme C appartient à Π, il existe une substitution close θ telle que $I(C\theta)=F$ et $C\theta$ est une clause minimale dans $GR(\Sigma)$. Par monotonicité de <, nous avons $(l=r)\sigma\theta < C\theta$. Cependant nous ne pouvons pass avoir $I((l=r)\sigma\theta)=F$ car I est un noeud d'échec de S*. Donc:

$$I((l=r)\sigma\theta)=V$$

Par conséquent, puisque $C\theta$ admet unlittéral vrai dans l'interprétation I, nous avons aussi:

$$I(C\theta)=V$$

Ceci est en contradiction avec l'hypothèse selon laquelle $C\theta$ est falsifiée par le noeud d'échec I.

La règle de subsomption fonctionnelle peut se généraliser en la règle de:

Subsomption de Preuve Equationnelle:
Si une clause C contient un littéral $f(s_1,\ldots,s_n) = f(t_1,\ldots,t_n)$ tel que pour tout i, $E|=(s_i=t_i)$, où E représente l'ensemble des équations de S, alors on peut supprimer C de l'ensemble des clauses.

CHAPITRE 5

Stratégies de superposition.

Dans ce chapitre, nous étudions un système d'inférence complet pour la réfutation qui étend la procédure de Knuth et Bendix . Dans le la cadre de la logique clausale du premier ordre avec égalité, ce système a été proposé par Lankford (1975) sous le nom de "derived reduction algorithm". Voir également (Brown 1974). Il admet essentiellement deux règles d'inférence: la résolution et la superposition généralisée. Cette dernière est une version très raffinée de la paramodulation. Admettons que certaines littéraux équationnels (positifs) sont orientés par un ordre de réduction. Alors un sous-terme d'une clause différent d'une variable peut être remplacé par un égal si (i) il est filtré par le membre gauche d'un littéral équationnel positif et (ii) s'il est sous-terme d'un littéral équationnel positif orienté, alors il doit appartenir au membre gauche de ce littéral. Comme dans la paramodulation classique, les autres littéraux des clauses parentes sont ajoutés à la clause résultat, après instanciation. Nous prouvons que cette stratégie est complète lorsque l'ordre de réduction utilisé est total sur l'ensemble des termes clos. Nous montrons également que la complétude est préservée en présence des règles simplification et de subsomption.

Lorsque toutes les clauses engendrées par le système sont des équations orientables, la stratégie précédente se réduit à la procédure de Knuth et Bendix. Cependant, même lorsque certaines équations ne sont pas orientables, il arrive que l'algorithme termine sur un ensemble fini d'équations dont toutes les instances closes sont orientables. Nous démontrons que ce système final est encore canonique, ce qui généralise la preuve de correction de l'algorithme de Knuth et Bendix due à Huet (1981), toujours sous réserve que l'ordre utilisé puisse être étendu en ordre total sur les termes clos. Un résultat analogue a été obtenu par (Bachmair et al. 1987)(voir aussi Bachmair 1987) par la méthode des *ordres bien-fondés sur les preuves*.

Lorsque les données sont des clauses de Horn, la stratégie de superposition généralisée s'apparente à une procédure de complétion conditionnelle. Si partant d'un ensemble consistant, l'application itérée des règles d'inférences conduit à un système fini, ce dernier peut être utilisé pour résoudre le problème du mot par normalisation (dans la théorie axiomatisée par l'ensemble initial). A la différence de la réécriture conditionnelle "classique", nous autorisons les paramodulations dans les conditions d'une règles lorsque celles ci ne sont pas plus petites que la conclusion. Cette technique nous a permis de compléter de nouveaux systèmes d'axiomes.

D.Lankford a démontré la complétude de la paramodulation généralisée dans le cas où les littéraux équationnels positifs n'apparaissent que dans des clauses unitaires et l'ensemble initial d'équations est canonique. Une procédure analogue est décrite dans (Fribourg 1985); celle-ci autorise n'importe quel ordre pour orienter les équations; cependant la complétude n'est obtenue qu'au prix des axiomes de réflexivité fonctionnelle et de la paramodulation dans les variables. Par ailleurs, l'auteur ne montre pas que cette complétude est préservée en présence de règles de subsomption et de simplification. E.Paul (1986) a étudié le cas des clauses de Horn: cependant sa procédure échoue comme la procédure de Knuth et Bendix lorsqu'une équation non-orientable se présente. Par ailleurs le type de superposition qu'il utilise est moins raffiné puisqu'il peut paramoduler dans les sous-termes de membres droits d'équations apparaissant dans des clauses non unitaires. Cette dernière remarque s'applique également à la stratégie proposée par (Bachmair et al. 1987) qui est une variante complète de celle de E.Paul.

SECTION 1: UNE STRATEGIE DE SUPERPOSITION

Introduction.

Le point de départ de ce travail est la remarque suivante de (Peterson 1983): "...,no one has developed a refutation complete set of inference rules for all of first-order logic with equality which reduces to the Knuth-Bendix procedure when restricted to equality units.". Nous présentons ces règles d'inférence et montrons leur complétude dans le cas où un ordre de simplification complet est utilisé pour comparer les termes.

Une procédure de démonstration automatique est décrite dans (Lankford 1975)(voir aussi Fribourg 1985) (appelée *derived reduction algorithm*) comportant essentiellement deux règles d'inférence: la résolution et la superposition généralisée. Cette dernière est une version très restrictive de la paramodulation. Etant donné un ordre de réduction sur les termes, on suppose certaines équations orientées de la gauche vers la droite (du membre le plus complexe au moins complexe).

Un sous-terme non-variable *s* d'une clause C peut être remplacé par un égal seulement si:

condition a: s est filtré par le membre gauche d'un littéral équationnel positif.

condition b: si s est un sous-terme d'un littéral équationnel positif de C, alors il appartient au membre gauche de cette équation.

Les autres littéraux des clauses parentes sont simplement ajoutés à la clause obtenue comme dans la version classique de la paramodulation.

Dans la suite nous montrons la complétude de cette stratégie, même lorsque l'on ajoute des règles réductrices comme la démodulation, la subsomption et la règle d'élimination des tautologies.

Quand toutes les clauses sont des équations orientables, la stratégie précédente se réduit à l'algorithme de Knuth-Bendix. Notre résultat est donc une extension des procédures de complétion sans échec de (Hsiang Rusinowitch 1987) ou (Bachmair Dershowitz Plaisted 1987) au calcul des prédicats du premier ordre avec égalité.

Nous soulignons que cette procédure n'utilise pas les axiomes de réflexivité fonctionnelle, et n'applique jamais la paramodulation dans des variables (paramodulation speciale). Ces restrictions sont essentielles pour l'efficacité d'un démonstrateur construit sur la règle de paramodulation. Lankford a prouvé la complétude de cette stratégie dans le cas particulier où le symbole d'égalité n'apparaît pas dans un littéral positif d'une clause non unitaire et de plus l'ensemble initial des équations forme un système canonique.

Paul (Paul 1985) a étudié le cas des clauses de Horn: cependant son algorithme échoue tout comme l'algorithme de Knuth-Bendix, lorsqu'une équation non-orientable apparait. Sa stratégie admet aussi un espace de recherche plus grand car elle n'empêche pas de remplacer des termes dans les membres gauches des équations qui apparaissent dans des clauses non-unitaires. Cette dernière remarque est encore valable pour la stratégie unitaires sur les clauses de Horn proposée par (Bachmair Dershowitz Plaisted 1987).

Une procédure analogue décrite dans (Fribourg 1985) permet n'importe quelle orientation des équations (pas seulement des ordres de réduction). Cependant les axiomes de réflexivité fonctionnelle, et la paramodulation dans les variables sont nécessaires pour démontrer la complétude

de la méthode. L.Fribourg ne montre pas non plus que la complétude est préservée lorsqu'on ajoute des règles de réduction.

Notre preuve de complétude repose sur la méthode des arbres sémantiques transfinis (comme au Chapitre 2) et aussi sur une extension de la notion de noeud d'échec que nous appelons noeud de quasi-échec. Un noeud de quasi-échec pour un ensemble de clauses S est une interprétation I qui falsifie une clause obtenue en simplifiant un élément de S par les équations orientées valides de I.

1. Règles d'inférence.

Introduisons maintenant les règles d'inférence de la stratégie de superposition. Comme d'habitude, nous supposons que $>$ est un ordre de simplification complet.

O-Factorisation

Si L_1 ,...,L_k sont des littéraux d'une clause C qui sont unifiables avec pour unificateur principal σ, et si pour tout atome A de $C-\{L_1,L_2,...,L_k\}$, $L_1\sigma \not\$ A\sigma$, alors $D=C-\{L_2\sigma,...,L_k\sigma\}$ est un *O-facteur* de C.

O-Résolution

Soit $C_1=L_1 \vee D_1$ et $C_2=\neg L_2 \vee D_2$ deux clauses telles que L_1 et L_2 soient unifiables, d'unificateur principal σ. Supposons de plus que, $L_1\sigma \not\$ A\sigma$ pour tout atome A de D_1 et $L_2\sigma \not\$ A\sigma$ pour tout atome A de D_2, alors $D=D_1\sigma \vee D_2\sigma$ est un *O-résolvant* de C_1 et C_2.

Paramodulation Orientée:

Soit $C_1=(s=t) \vee D_1$ et $C_2 = C_2[n\leftarrow r]$, où r est un sous-terme non-variable à l'occurrence n du littéral L de C_2. Si

(1) $s\sigma=r\sigma$ avec σ unificateur principal de s et r,
(2) $s\sigma \not\$ t\sigma$,
(3) L n'est pas une équation.
(4) $L\sigma \not\$ A\sigma$, $\forall A \in C_2 - \{L\}$,

alors $C = C_2[n\leftarrow t]\sigma \vee D_1\sigma$ est un *paramodulant orienté de C_1 dans C_2 à l'occurrence n.*

Superposition Généralisée:

Soit C_1 une clause (s=t) $\vee D_1$. Soit C_2 une autre clause et a=b un littéral de C_2. Soit r un sous-terme non-variable de a à l'occurrence n de C_2 tel que:

(1) $s\sigma = r\sigma$ avec σ unificateur principal de s et r,
(2) $s\sigma \not\$ t\sigma$
(3) $a\sigma \not\$ b\sigma$
(4) $(a=b)\sigma \not\$ A\sigma$, $\forall A \in C_2 - \{a=b\}$,

alors $C = C_2[n\leftarrow t]\sigma \vee D_1\sigma$ est un *superposant généralisé de C_1 à l'occurrence n* de C_2.

Remarque:

Lorsque C1 et C2 sont des règles de réécriture, un superposant généralisé de C1 dans C2 n'est autre qu'une superposition au sens de Knuth et Bendix. Dans le cas où toute clause est soit une égalité soit une inégalité, les seules règles applicables sont la superposition étendue et la résolution avec x=x. La stratégie que nous obtenons alors est la S-stratégie de (Hsiang Rusinowitch 87). De plus, quand il n'y a pas d'inégalité dans le système et chaque égalité est orientable, la procédure coincide exactement avec l' algorithme de complétion de Knuth et Bendix.

Nous ajouterons une légère restriction sur les ordres de simplification complets utilisables dans la définition des règles d'inférence; nous supposerons désormais qu'ils satisfont la propriété suivante, qui signifie que toutes les égalités closes de membre gauche identique sont adjacentes:

O6: si $(u=w) < A < (u=v)$, $u>w$ et $u>v$, où u,v et w sont des termes clos, alors il existe un terme clos t tel que A est égal à l' atome $(u=t)$.

Soulignons que cette condition est satisfaite par tous les CSO's considérés dans les Chapitres précédents .

2.Exemples

2.1.Exemple

L' exemple simple qui suit illustre nos règles d'inférence. Il montre la transitivité de INF en supposant l'associativité de max. La négation skolémisée du théorème est la conjonction des clauses 5,6,7. Nous utiliserons un ordre qui compare d'abord les prédicats comme dans la section 2.2.3 du Chapitre 1, avec la précédence suivante sur les symboles de fonctions: max>a>b>c.

1. INF(x,y) $\vee$ INF(y,x).
2. $\neg$INF(x,y) $\vee$ max(x,y) $\rightarrow$ y.
3. $\neg$INF(y,x) $\vee$ max(x,y) $\rightarrow$ x.
4. max(max(x,y),z) $\rightarrow$ max(x, max(y,z)).
5. INF(a,b).
6. INF(b,c).
7. $\neg$INF(a,c).

8. max(a,b) $\rightarrow$ b par résolution de 5,2.
9. max(b,c) $\rightarrow$ c par résolution de 6,3.
10. INF(x,y) $\vee$ max(x,y) $\rightarrow$ x. par résolution de 1,3.
11. max(a,c) $\rightarrow$ a par résolution de 7,10.
12. max(a,max(b,z)) $\rightarrow$ max(b,z) par superposition de 8 dans 4.
13. max(a,c) $\rightarrow$ max(b,c) par superposition de 9 dans 12.
14. max(b,c) $\rightarrow$ a par superposition de 11 dans 13.
15. a $\rightarrow$ c par superposition de 9 dans 14.
16. $\neg$INF(c,c) par paramodulation de 15 dans 7.
17. INF(x,x) par factorisation de 1.
18. [] par résolution de 16 et 17.

2.2.Exemple

Donnons maintenant un exemple plus complexe, emprunté à (Brown 1974). Il montre que le quotient des carrés de deux nombres premiers entre eux n'est pas premier . Bien sûr, il existe un résultat plus general , mais sa preuve nécessite un raisonnement par induction. Nous utilisons la précédence suivante sur les symboles de fonction: /*>.>+>b>c>a*; le status des opérateurs binaires est gauche-droite.

Axiomes pour l'addition et la multiplication:

1. (x+y)+z=x+(y+z).
2. x+y=y+x.
3. o+x=x
4. x+(-x)=o.
5. (x.y).z=x.(y.z).
6. x.y=y.x.
7. w.(x+y)=w.x+w.y.
8. (-x).y=-(x.y).
9. x.y≠o V x=o V y=o.

Propriétés des prédicats "DIVISE" et "PREMIER" :

D1. ¬D(x,y) V (y|x).x=y.
D2. D(x,y) V (y|x).x ≠y.
D3. ¬P(o)
D4. ¬P(1).
D5. ¬P(-1).
D6. ¬P(z) V ¬D(x,z) V x=1 V x=-1 V x=z V x=-z.
D7. ¬P(z) V ¬D(z,x.y) V D(z,x) V D(z,y).
D8. (x.y)|y=x V y=o.

Négation du théorème:

H1. P(c).
H2. (b.b).c=a.a.
H3. ¬D(z,a) V ¬D(z,b) V z=1 V z=-1.

P1. (a.a)|c=b.b V c=o par par. de H2 dans D8.
P2. ¬D(c,x.y) V D(c,x) V D(c,y). par res. de D7,H1
P3. ((x.y)|c).c ≠ x.y) V D(c,x) V D(c,y) par res. de P2,D2
P4. ((x.x)|c).c ≠ x.x) V D(c,x) par fact de P3
P5. ((x.x)|c).c ≠x.x V (x|c).c=x par res. de D1,P4
P6. (b.b).c ≠ a.a V (a|c).c=a V c=o par par. de P1 dans P5.
P7. (a|c).c = a V c=o par par. de H2 dans P6 (et res. avec x=x)
P8. c.(a|c) = a V c=o par par. de 6 dans P7 (status de . est l-r)
P9. c.((a|c).z)=a.z V c=o. par super de 5 et P8
P10. z.x+z.y ≠ o V z=o V x+y=o par par. de 7 dans 9
P11. c.x+a.z ≠o V c=o V x+(a|c).z=o. par par. de P9 dans P10

P12. x.c+a.z ≠o V c=o V x+(a|c).z=o par par. de 6 dans P11.
P13. a.a+a.z ≠o V c=o V b.b+(a|c).z=o par par. de H2 dans P12
P14. a.a+(-(a.w))≠o V c=o V b.b+(-(a|c).w)=o par par. de 6 et 8 dans P13
P15. a.a+(-(a.w))≠o V c=o V b.b+((-(a|c).w)+z)=z par super de 1 et P14.
et reduction par 3.
P16. a.a+(-(a.w))≠o V c=o V b.b=(a|c).w par super de 2,4 et P15.
P17. c=o V (a|c).a=b.b par par. de 2,4 dans P16.
P18. D(z,x.z) V x.z≠x.z V z=o par par. de D8 dans D2.
P19. D(c,x.a) V c=o par par. de P9 dans P18
P20. D(c,b.b) V c=o par par. de P17 dans P19
P21. ¬P(c)V D(c,b) V c=o par res. de D7 et P20
P22. D(c,b) V c=o par res. de H1 et P21
P23. ¬D(c,a) V c=o V c=1 V c=-1 par res. de P22 et H3
P24. (a|c).c ≠a V c=o V c=1 V c=-1 par res. de P23 et D2.
P25. c=o V c=1 V c=-1 par par. de P7 dans P24 et res. avec x=x
P26. P(o) V P(1) V P(-1) par successive par. de P26 dans H1.
P27. [] par successives res. de P27 et D3 D4 D5.

Montrons maintenant que la paramodulation orientée, la superposition généralisée, la o-factorisation et la o-résolution forment une stratégie de réfutation complète.

Nous appellerons cette stratégie *RP*.

3. Théorème de Complétude.

Théorème 3.1: Soit S un ensemble E-inconsistant de clauses contenant x=x alors S* contient la clause vide.

Preuve:

La preuve que nous donnons est analogue à la preuve de complétude de l'algorithme de complétion sans échec de Knuth-Bendix-Huet (Hsiang Rusinowitch 87). Cependant, comme nous considérons maintenant des clauses multi-littérales, de nouvelles difficultés apparaissent. L'essentiel de notre effort sera consacré aux clauses du type:

s=a V s=b V....

Supposons, par l'absurde, que RP ne soit pas une stratégie complète; alors il existe un ensemble E-inconsistant de clauses S tel que MCT(S*) n'est pas vide. Nous définissons une suite de noeuds dans MCT(S*) par induction transfinie et montrons que la branche obtenue est vide, en contradiction avec les hypothèses.

Introduisons la notion de *noeud de quasi-échec*: c'est une interprétation R qui falsifie une clause obtenue en simplifiant un élément de S*, avec les équations valides de R:

Définitions 3.2:

Soit R un noeud de MCT(S*) défini sur W(B+1). Ce noeud R est un *noeud de quasi-échec* (pour S) si:

1. R(B)=F
2. B is une égalité s=t (avec s>t)
3. il existe une instance close D d'une clause C de S telle que tous les atomes de D sont strictement plus petits que s=s, et telle qu'il existe une clause D' vérifiant R(D')=F et D→*(R) D'.

Nous dirons alors que la clause C *quasi-étiquette* le noeud R. Nous dirons aussi que D est *quasi-fausse* pour R.

Remarquons que lorsque 3. est vérifié pour une clause D', alors pour toute clause D" telle que D→*(R)D" et D" est dans le domaine de R, nous avons aussi R(D")=F. En effet R peut se prolonger en E-interprétation.

Soit o le plus petit atome de A(P,F). Nous construisons une suite Σ de E-interprétations indicée par l'ensemble bien ordonné A(P,F). Posons d'abord: $I_o = \emptyset$ (interprétation vide). Supposons maintenant que $I_{B"}$ soit défini pour tout B" de l'intervalle W(B').

Plusieurs situations sont à examiner pour définir l'élément d'indice B' de la suite:

Premier cas: B' n'est pas un ordinal limite.

Donc, B' a un prédécesseur dans A(P,F), que nous noterons B. Supposons que K' est le dernier élément de la suite, que nous ayons défini. Alors W(B) est le domaine de l'interprétation K'.

Plusieurs cas peuvent se présenter:

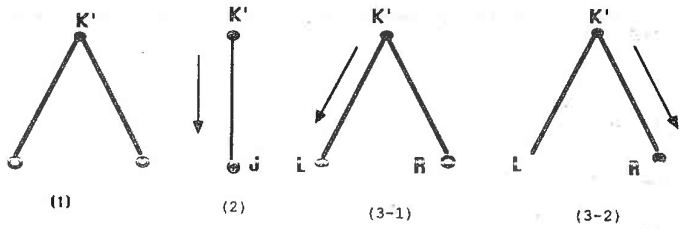
(1) si K' n'a aucun successeurs dans MCT(S*) alors la suite est achevée.

(2) si K' a exactement un successeur J dans ET et J appartient aussi à MCT(S*) alors c'est l'élément suivant de la suite.

(3) si K' a deux successeurs L et R dans ET avec L(B)=T et R(B)=F alors

(3.1) si R est un noeud de quasi-échec ou un noeud d'échec et L est dans MCT(S*) alors l'élément suivant est L.

(3.2) si R n'est ni un noeud d'échec ni un noeud de quasi-échec, c'est l'élément suivant.



Second cas : B' est un ordinal limite.

Nous définissons simplement $I_{B'}$ comme la limite de $I_{B"}$ quand B" tend vers B'.Par conséquent, dans ce cas il est toujours possible de définir l'élément d'indice B'.

La suite de noeuds Σ n'est pas vide car MCT(S*) est supposé non vide. Mais elle ne peut être définie pour tous les indices de A(P,F). Sinon sa limite (inductive) serait définie sur A(P,F) et constituerait un modèle égalitaire pour S, qui est par hypothèse E -inconsistant.

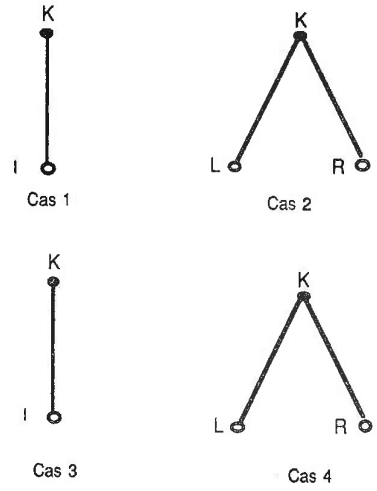
Soit B le plus petit atome pour lequel I_B n'est pas défini. Nous avons vu dans le second cas ci-dessus que, lorsque B' est un ordinal limite et la suite est défini sur W(B'), nous pouvons toujours définir $I_{B'}$. Donc B n'est pas un ordinal limite et, par conséquent la suite termine toujours de l'une des manières décrites ci-dessous, où K est le dernier élément de la suite, dont le domaine est W(B):

cas 1: K possède exactement un successeur I qui est un noeud d'échec et B est K irréductible.

cas 2: K a deux successeurs, L et R qui sont des noeud d'échecs.

cas 3: K possède exactement un successeur I qui est un noeud d'échec et B est K-réductible.

cas 4: K a deux successeurs, L et R (L(B)=T et R(B)=F), avec L noeud d'échec et R noeud de quasi-échec.



Nous prouverons dans tous les cas qu'il existe une clause de S* qui est fausse dans l'interprétation K, en contradiction avec l'hypothèse selon laquelle K appartient à MCT(S*).

Cas 1 et 2: Ces cas ont été traités au chapitre 2. Une étape de résolution sur les clauses de S* qui sont falsifiées par les successeurs de K engendre une clause de S* qui est falsifiée par K.

Avant de considérer les autres cas, nous allons démontrer quelques lemmes techniques précisant la structure des clauses qui (quasi-)étiquettent les (quasi-)noeuds d'échecs.

Lemme A:

Soit K' un noeud de la suite Σ, qui admet deux successeurs L et R, tels que W(K')= [o,(s=t)[et tel que R (celui de droite) est un noeud d'échec ou un noeud de quasi-échec. Alors, pour toute clause C qui (quasi-)étiquette R, et pour toute instance close D de C qui est (quasi-)fausse pour R, il n'existe aucun terme u tel que s≠u soit un littéral de D.

preuve: Soit D' tel que R(D')=F et D->*(K')D'. Supposons que (s=u) est un atome de D, et (s'=u') est l'atome de D' qui vérifies: (s=u)->*(K')(s'=u'). Remarquons d'abord que nous ne pouvons pas avoir s' différent de s, sinon s=t serait réductible par une égalité de K': ce cas est exclus car K' a deux successeurs. Ainsi s est identique à s', et, comme (s=u') ≤ (s=t), nous avons aussi u'≤t. Montrons maintenant que R(s=u')=F. Si u' est le terme t, R(s=u')=F car R est le successeur droit de K'. Si u'<t, cela vient du fait que s=t est K'-irréductible et donc s=u' ne peut servir à K'-réduire s=t. Comme R(D')=F, R(s'=u') étant égal à F, nécessairement s'=u' est un littéral positif de D'. Donc le littéral correspondant de D doit être aussi positif: en conclusion s=u est un littéral positif de D. Ceci achève la preuve du Lemme A.

Ainsi D peut s'écrire:

(*) $s=u_1 \vee s=u_2 \vee ... \vee s=u_m \vee s=u_{m+1} \vee ... \vee s=u_k \vee D''$

avec $s=u_i \rightarrow^*(K') s=t$ pour $1 \le i \le m$ et

$s=u_i \rightarrow^*(K') s=v_i$ pour $m<i \le k$ et

s n'est pas un sous-terme de D" .

Avant de continuer, nous remarquons que chaque littéral de D" est strictement plus petit que toute égalité avec s dans un membre.Sinon si L∈ D" vérifie L>s=u, alors d'après l'hypothèse selon laquelle L<s=s (rappelons que K' est un (quasi-)noeud d'échec), nous obtenons une contradiction avec l'hypothèse O6.

Lemme AA : sous les hypothèses du Lemme A, il existe un i tel que $s=u_i \rightarrow^*(K')s=t$.

Le Lemme signifie que m est différent de 0 dans l'expression de D. Prouvons le. Si m=0 , soit s=v la K'-forme normale maximale des atomes $s=u_i$. Nous notons que v<t et K'(s=v)=F.

Soit K" la restriction de K' au domaine W(s=v) et R" le successeur droit de K" dans ET. Comme chaque égalité utilisée pour K'-réduire D est strictement plus petite que s=v, nous avons aussi D→*(K") X où X est une clause vérifiant R(X)=F et X est K'-irréductible. Comme chaque K'-forme normale des atomes de D est plus petite où égale à s=v (cf. la remarque avant le Lemme AA), nous avons aussi R"(X)=F. Nous avons ainsi démontré que K" vérifie la condition 3.1. Cependant, ceci est impossible car R" appartient à la suite Σ (comme restriction de K').

Ces lemmes nous permettent de discuter les autres cas.

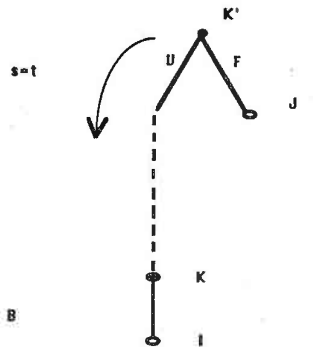
Cas 3

Nous savons que I, le successeur du dernier noeud K de Σ, est un noeud d'échec, qui falsifie une instance close d'une clause C de S*.

sous-cas 3.1 : B n'est pas un atome d'égalité.

Soit s=t un atome d'égalité I-irréductible tel que s>t, s est un sous-terme de B, et K(s=t)=T. Il existe un élément de ce type car B est K-réductible; il est donc possible d'appliquer le théorème de réduction du Chapitre 2.

Soit K' la restriction de K au domaine W(s=t), et soit J le successeur droit de K' dans ET. comme J n'est pas dans la suite Σ que nous avons construite, K' vérifie la condition 3-1. Par conséquent, il existe une instance close D d'une clause de S* tel que D→*(K')D', J(D') = F et tout atome de D est strictement plus petit que s=s. Nous pouvons choisir D minimale (pour l'ordre <<, qui est, par définition, l'extension multi-ensemble de <).



Si nous appliquons le Lemme A avec K', J et D nous pouvons écrire D:

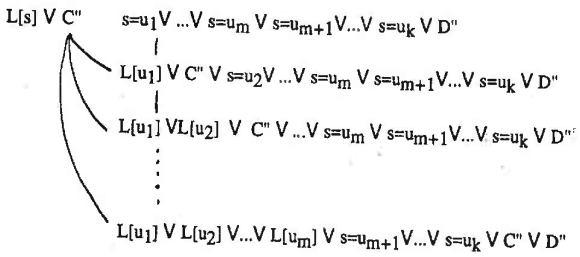
$$s=u_1 \vee s=u_2 \vee ... \vee s=u_m \vee s=u_{m+1} \vee ... \vee s=u_k \vee D''$$

Remarquons que C s'écrit L[s] ∨ C", où L[s] représente soit le littéral B soit le littéral ¬B.

Après quelques étapes de paramodulation orientée et de factorisation, nous obtenons la clause P (qui appartient, par définition à S*):

$$L[u_1] \vee L[u_2] \vee ... \vee L[u_m] \vee s=u_{m+1} \vee ... \vee s=u_k \vee C'' \vee D''$$

La déduction de P est détaillée dans l'arbre suivant:



Nous avons construit un paramodulant dont tous les littéraux sont faux dans l'interprétation K:

Lemme B: K(P)=F

Preuve:
K(C")=F comme nous avons K(C)=F.

Chacun des atomes de D" est strictement plus petit que s=t; par conséquent, D" est dans le domaine de K' et K'(D")=F. Mais K est une extension de K'; donc, K(D")=F.

Pour i>m nous avons $s=u_i \rightarrow^*(K')$ $s=v_i$ et K'(s=v_i)=F. Chaque atome d'égalité valide pour K' est aussi valide pour K.

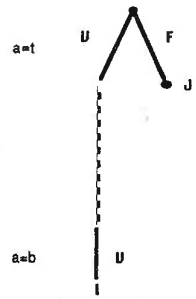
Nous pouvons donc remplacer K' ci-dessus par K. Mais $s=u_i$ est dans le domaine de K car il est strictement plus petit que L[s]. Nous en déduisons l'égalité K(s=u_i)=F.

Pour 1≤i≤m nous avons $s=u_i \rightarrow^*(K')$ s=t. Donc K(s=u_i)=V et I(L[u_i]) = I(L[s]) = F. Mais L[u_i] < L[s], donc chaque littéral de P est dans le domaine de K. Comme I est une extension de K, par consistance, nous avons K(L[u_i])=F. Le Lemme est démontré.

sous-cas 3.2: B est un atome d'égalité "a=b" avec a>b et I(B)=T.

Soit s=t l'égalité minimale telle que I(s=t)=T et s est un sous-terme de B. Si s est un sous-terme strict de a ou s est un sous-terme de b alors nous procédons comme avant.

Montrons maintenant que s ne peut pas être égal à a. Si c'était le cas, de I(a=t)=V et I(a=b)=V nous déduirions I(b=t)=V. Comme b>t, b=t permet de I-réduire B; c'est impossible, car aucune égalité plus petite que s=t ne peut I-réduire B.



sous-cas 3.3: B est un atome d'égalité "a=b" avec a>b et I(B)=F.

S'il existe un sous-terme strict *a'* de *a* tel que I(a'=b')=V pour un b'≤ a', il suffit de reprendre la preuve du sous-cas 3.1.

Nous supposerons désormais qu'un tel a' n'existe pas.

sous-cas 3.3.1: pour tout d≤b nous avons I(a=d)=F.

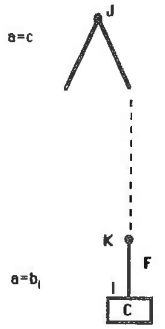
L'hypothèse 3.3.1 implique que tout atome de C du type a=d apparait seulement dans un littéral positif. Donc, C peut s'écrire a=b_1 V a=b_2 V...V a=b_m V C" et *a* n'a pas d'occurrence dans la sous-clause C".

Nous utiliserons l'hypothèse (O6) sur l'ordre < :

si (u=v) > A > (u=w) alors il existe un terme clos x tel que A est égal à u=x.

Cette hypothèse nous permet de reformuler l'affirmation précédente: chaque atome *a=d* de C est strictement plus grand que n'importe quel atome de C".

Soit a=c la I-forme normale maximale (pour <) des atomes a=b_i où i≤m; autrement dit : c est sup{k_i : 1≤i≤m} où k_i=inf{k :I(k=b_i)=V}.



Soit J la restriction de I à W(a=c). Tout égalité utilisée pour I-réduire un des b_i est nécessairement strictement plus petite que a=c.

En effet, il n'y a pas d' égalité a=z telle que I(a=z)=V (hypothèse 3.3.1) et il n'y a pas d'égalité s'=t' telle que s' est un sous-terme strict de *a* et I(s'=t')=V (hypothèse 3.3), par conséquent nous pouvons I-réduire un atome a=b_i seulement avec une égalité dont le membre gauche est un sous-terme de b_i; une égalité de ce type est toujours plus petite que a=c. Ces remarques assurent l'existence d'une clause close C! telle que C→*(J) C! et chaque littéral de C! est inférieur à (a=c). Donc, J vérifie la condition 3.1. Aussi, le successeur de J ne peut appartenir à la suite de noeuds que nous avons définie dans MCT(S*). Mais comme K(a=c)=F, K ne suit pas J dans Σ. Donc le sous-cas 3.3.1 est impossible.

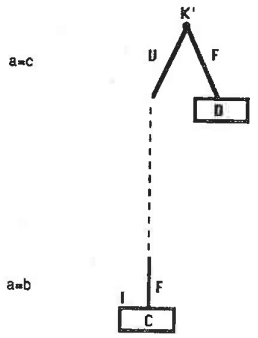
Sous l'hypothèse 3.3 nous aurons toujours :

sous-cas 3.3.2: il existe un terme c tel que c<b et I(a=c)=V.

Prenons pour c le plus petit terme vérifiant 3.3.2.

Soit K' la restriction de I au domaine W(a=c), et soit J le successeur droit de K' dans ET. Comme J n'appartient pas à la suite Σ, K' vérifie la condition 3-1. Il existe donc une instance close D d'une clause de S* telle que D→*(K')D', J(D')=F et tout atome de D est strictement plus petit que a=a . Nous pouvons supposer que D est minimale (pour <<).

D'après le Lemme A, et le Lemme AA, toute égalité a=u ayant une occurrence dans la clause D apparait dans un littéral positif.



Ainsi D peut s'écrire:

(*) $a=u_1 \vee a=u_2 \vee ... \vee a=u_m \vee a=u_{m+1} \vee ... \vee a=u_k \vee D''$

avec $a=u_i \rightarrow^*(K')\ a=c$ pour $1 \le i \le m$ et

$a=u_i \rightarrow^*(K')\ a=v_i$ pour $m<i \le k$ et

a n'est pas un sous-terme de D" .

Nous savons que I est un noeud d'échec, qui falsifie une instance C d'une clause de S*.

L'hypothèse entraine que C (ou un facteur de C) peut s'écrire a=b ∨ C" où tous les atomes de C" sont strictement plus petits que a=b.

Supposons d'abord que C>D.

Quelques étapes de superposition généralisée (et factorisation) avec C et D comme clauses initiales engendrent la nouvelle clause P :

$[u_1=b] \vee [u_2=b] \vee ... \vee [u_m=b] \vee a=u_{m+1} \vee ... \vee a=u_k \vee C'' \vee D''$

L'arbre de déduction est identique à celui du sous-cas 3.1.

Pour $m+1 \le i \le k$, $a=u_i \rightarrow^*(I)\ a=v_i$ car I est une extension de K'. Mais $I(a=v_i)=F$. Mais $a=u_i$ est dans le domaine de I, car $a=u_i<a=b$ d'après l'hypothèse "C>D". Donc nous avons aussi $I(a=u_i)=F$.

Pour $i<m+1$, $a=u_i \rightarrow^*(I)\ a=c$ car I est une extension de K'. Donc $I(u_i=c)=V$. Mais I(a=c)=V. Par conséquent nous avons aussi $I(u_i=a)=V$. Puis de I(a=b)=F nous déduisons $I(u_i=b)=F$. Comme dans le Lemme B, il est facile de voir que I(C"∨D")=F pour conclure avec I(P)=F. Comme chaque atome de C est strictement plus petit que a=b, nous avons aussi K(P)=F. Cela signifie que K est un noeud d'échec: c'est contradictoire avec le fait que K appartient à MCT(S*).

Supposons maintenant que C≤D:

Pour distinguer les atomes contenant le sous-terme a, nous écrirons C sous la forme suivante : $L_1[a] \vee L_2[a] \vee ... \vee L_r[a] \vee C''$ où $L_h[a]$ est soit $a=b_h$ soit $a \neq b_h$ et *a* n'a pas d'occurrence dans C". Nous supposerons aussi que $L_1[a] > L_2[a] >, ..., > L_r[a]$.

De chaque égalité $a=u_i$ ($i \le m$) de D nous paramodulons ou nous superposons successivement dans chaque occurrence de a dans C pour obtenir une nouvelle clause LP:

$L_1[u1] \vee L_1[u_2] \vee ... \vee L_1[u_m] \vee$
$L_2[u1] \vee L_2[u_2] \vee ... \vee L_2[u_m] \vee$

.....

$L_r[u1] \vee L_r[u_2] \vee ... \vee L_r[u_m] \vee$

$a=u_{m+1} \vee ... \vee a=u_k \vee C'' \vee D''$

qui peut s'obtenir par la déduction suivante:

$L_1[a] \vee L_2[a] \vee ... \vee L_r[a] \vee C''$

$a=u_1 \vee ... \vee a=u_k \vee D''$

$L_1[u_1] \vee L_2[u_1] \vee ... \vee L_r[u_1] \vee C'' \vee a=u_2 \vee ... \vee a=u_k \vee D''$

$L_1[u_1] \vee L_2[u_1] \vee ... \vee L_r[u_1] \vee$
$L_1[u_2] \vee L_2[u_2] \vee ... \vee L_r[u_2] \vee C'' \vee a=u_3 \vee ... \vee a=u_k \vee D''$

.....

$L_1[u_1] \vee L_2[u_1] \vee ... \vee L_r[u_1] \vee$
$L_1[u_2] \vee L_2[u_2] \vee ... \vee L_r[u_2] \vee$

.....

$L_1[u_m] \vee L_2[u_m] \vee ... \vee L_r[u_m] \vee C'' \vee a=u_m \vee ... \vee a=u_k \vee D''$

Lemme C: il existe une clause LP' tel que $LP \rightarrow^*(K') LP'$ et J(LP')=F.

Preuve:

Pour $i<m+1$, $I(u_i=c)=V$. Mais $I(a=c)=V$. Par conséquent nous avons aussi $I(u_i=a)=V$. Mais $I(L_j[a])=F$, pour $j \leq r$, entraine $I(L_j[u_i])=F$ et aussi $K'(L_j[u_i])=F$. Comme C'' et D'' sont plus petits que (a=c), la sous-clause $(a=u_{m+1} \vee ... \vee a=u_k \vee C'' \vee D'')$ peut être K'-réduite en une clause X telle que J(X)=F. Donc LP elle-même est K'-réductible en une clause falsifiée par J.

Comme m est différent de 0, LP est strictement plus petite que D. Nous obtenons une contradiction avec le fait que D est une clause minimale vérifiant la condition 3.1 au noeud K'.

Cas 4

Le dernier noeud K de la suite Σ a deux successeurs L et R: L est un noeud d'échec et R est un noeud de quasi-échec. Donc, il existe une égalité s=t telle que le domaine de K est W(s=t). Le Lemme A et le Lemme AA appliqués à K et R impliquent l' existence d'une clause D pouvant s'écrire:

(*) $s=u_1 \vee s=u_2 \vee ... \vee s=u_m \vee s=u_{m+1} \vee ... \vee s=u_k \vee D''$
avec $s=u_i \rightarrow^*(K') s=t$ pour $1 \leq i \leq m$ et
$s=u_i \rightarrow^*(K') s=v_i$ pour $m<i \leq k$ et
s n'est pas un sous-terme de D'' .

Comme L est un noeud d'échec (mais pas K), il existe une clause C pouvant s'écrire $s \neq t \vee C''$ qui est une instance close d'une clause de S* telle que $L(s \neq t \vee C'')=F$.

Pour éliminer certaines occurrences de s, nous appliquons une suite de paramodulations à partir de D et C, et nous obtenons la clause LP:

$t=u_1 \vee t=u_2 \vee ... \vee t=u_m \vee s=u_{m+1} \vee ... \vee s=u_k \vee C'' \vee D''$

Pour $i \leq m$ nous avons $K(u_i=t)=V$ (rappelons que $s=u_i \rightarrow^*(K) s=t$).

Après quelques factorisations éventuelles, nous pouvons supposer que C'' et D'' sont dans le domaine de K et vérifient K(C'')=K(D'')=F.

Pour i>m, $s=u_i \rightarrow^*(K) s=v_i$ avec $v_i<t$. Par conséquent, $LP \rightarrow^*(K) LP'$ et K(LP')=F. Comme chaque littéral de LP' est plus petit que s=t, nous avons également R(LP')=F. Cependant, aucun littéral de LP ne peut être K-réduit en s=t. Nous obtenons donc une contradiction avec le Lemme AA, car K appartient à S et vérifie la Condition 3.1 .

4. Lemmes de Relèvement.

Pour relever notre preuve du cas clos au cas variable, nous remarquons d'abord que chaque instance $C\theta$ d'une clause C de S* qui étiquette ou quasi-étiquette un noeud I, θ peut être choisie I-irréductible. Nous utilisons alors simplement les lemmes de relèvement pour la o-factorisation, la o-résolution, la o-paramodulation donnés au Chapitre 2. Pour relever la règle de superposition généralisée, l'argument est analoque à celui de la o-paramodulation (voir aussi le Lemme des Paires Critiques dans l'algorithme de Knuth et Bendix):

Lemme de Relèvement de la Superposition Généralisée:

Soit C1:: (s=t) V C et C2:: (a=b) V D deux clauses et n une occurrence non-variable de s. Soit SG:: $s\theta[n<-b\theta]=t\theta \vee C\theta \vee D\theta$ une superposition généralisée des instances closes $(s=t)\theta \vee C\theta$ et $(a=b)\theta \vee D\theta$ de C1 et C2. Alors il existe une superposition généralisée S de C1 et C2 telle que SG est une instance de S.

preuve: comme n est une occurrence de s, $s\theta/n = (s/n)\theta = a\theta$. Ainsi, s/n et a sont unifiables. Soit σ leur unificateur principal; alors il existe une substitution ϕ tel que $\theta=\sigma\phi$. De plus, nous avons $s\sigma \not\equiv t\sigma$ (sinon $s\sigma < t\sigma$ impliquerait $s\sigma\phi<t\sigma\phi$, en contradiction avec l'hypothèse selon laquelle $s\theta<t\theta$). De même as $\not\equiv$ bs. Donc nous pouvons construire une superposition généralisée S de C1 et C2 à l' occurrence n de s: $(s[n<-b] \vee C \vee D)\sigma$. Il suffit alors de vérifier que $S\phi=SG$.

5. Complétude en présence de règles de réduction.

Nous supposerons maintenant que les règles de réduction données au Chapitre 4 sont ajoutées à l'ensemble des règles d'inférence et que seules les dérivations équitables sont engendrées. La preuve de complétude s'étend exactement de la même manière qu'au Chapitre 4, une fois que nous avons démontré le lemme suivant:

Lemme 5.1: tout noeud de quasi-échec peut être quasi-étiqueté par une clause persistante.

Idée de la preuve:

Il suffit de reprendre la preuve du Chapitre 4 sur les noeuds d'échec en considérant l' ensemble

Σ' des clauses qui quasi-étiquettent I.

Nous montrons ensuite que lorsqu'une clause C qui quasi-étiquette I est simplifiée, nous obtenons une clause qui possède aussi cette propriété. Si une clause C' subsume C, alors elle quasi-étiquette aussi I.

6. Conclusion.

La stratégie décrite dans cette section se raffine lorsque nous traitons uniquement des clauses de Horn. Nous pouvons, par exemple, appliquer les paramodulations ou superpositions uniquement dans les littéraux maximaux de chaque clause. Cela sera détaillé dans la section 3 de ce chapitre. Il est aussi possible de montrer la complétude d'une stratégie unitaire comme au chapitre 3, section 2. Nous ne développerons pas ce point.

SECTION 2: PROBLEMES DE MOTS DANS LES THEORIES EQUATIONNELLES.

Introduction

La procédure de Knuth et Bendix permet de compléter les systèmes d'équations en ensembles canoniques de réductions, permettant de décider le problème du mot par simple normalisation. Rappelons que cette procédure consiste à trouver des paires critiques divergentes (i.e. dont les membres sont différents après normalisation), à les orienter en règles, et à maintenir les termes complètement réduits par le système courant de règles. Nous abrègerons "procédure de complétion de Knuth et Bendix" en "procédure KB".

La procédure KB échoue

- lorsqu'elle engendre une paire critique non-orientable ou
- lorsqu'elle engendre une infinité de règles.

Le premier problème est en général indécidable (Huet Lankford 1978). Pour certaines théories, comme la commutativité, il peut se résoudre en incorporant un algorithme spécial d'unification dans la procédure de complétion (Lankford Ballantyne 1977) (Peterson Stickel 1981) (Jouannaud Kirchner 1986).

G.Huet (1981) a proposé une semi-solution pour le second problème. En effet, en corollaire de sa preuve de correction de la procédure KB, G.Huet montre qu'elle fournit aussi une procédure de semi-décision pour le problème du mot. Plus précisément, supposons que la procédure de complétion est équitable et que les paires critiques engendrées sont toujours orientables (pour le même ordre de réduction), alors $s=t$ est valide dans la théorie considérée si et seulement si il existe une étape où s et t se réduisent au même terme.

Nous allons montrer que la restriction des règles d'inférences de la section 1 à des ensembles formés uniquement d'équations permet de construire une procédure de complétion sans échec, que nous appelons UKB; cette procédure ne s'arrête pas en échec lorsqu'elle rencontre une équation non-orientable, et présente les mêmes fonctionnalités que la procédure KB:

- si la procédure UKB s'arrête, elle fournit un système canonique,
- sinon elle fournit un algorithme de semi-décision pour le problème du mot.

1. Une procédure de complétion sans échec.

Nous supposerons que l'ensemble des prédicats est réduits au seul élément = et que toutes les clauses sont unitaires. Dans la suite, < est un ordre de simplification complet.

Exprimons les règles d'inférence de la section 1 dans ce cas particulier :

(la règle de factorisation n'a plus de sens pour des clauses unitaires).

Surréduction Généralisée (= Paramodulation Orientée):
Soit deux clauses s=t et g[r]≠d, où *r* est un sous-terme non-variable à l'occurrence *n* dans g≠d. Si
(1) $s\sigma = r\sigma$ avec σ unificateur principal de s et r,
(2) $s\sigma \not\downarrow t\sigma$,
alors $(g[t]\neq d)\sigma$ est une *surréduction généralisée de s=t dans g≠d à l'occurrence n.*

Superposition Généralisée:
Soit deux clauses s=t et a[r]=b où r est un sous-terme non-variable de a à l'occurrence n. Si :
(1) $s\sigma = r\sigma$, σ étant l'unificateur principal de s et r,
(2) $s\sigma \not\downarrow t\sigma$
(3) $a\sigma \not\downarrow b\sigma$
$(a[r]=b)\sigma$ est une *paire critique généralisée de s=t à l'occurrence n de a=b.*

Remarque fondamentale: lorsque les équations s=t et a=b sont orientables, alors les conditions (2) et (3) ci-dessus deviennent:
(2') $s\sigma > t\sigma$
(3') $a\sigma > b\sigma$
L'équation engendrée $(a[r]=b)\sigma$ est une paire critique, exactement comme dans l'algorithme de Knuth et Bendix.

Nous allons maintenant montrer qu'il est possible d'économiser la règle de **Résolution**, en la remplaçant par la règle plus simple de **Réfutation Réflexive.**

Considérons deux clauses *s=t* et *a≠b* unifiables, d'unificateur principal σ. Elles admettent un O-résolvant qui est la clause vide. Il n'est pas restrictif de supposer que, dans cet exemple, $s \not\downarrow t$ et s est unifiable avec a, l'unificateur principal étant θ. Comme σ est aussi un unificateur de s et a, il existe une substitution κ telles que $\sigma=\theta\kappa$. Il est également possible d'appliquer la règle de surréduction généralisée sur les clauses *s=t* et *a≠b* pour obtenir la clause $t\theta\neq b\theta$. Comme $t\theta\kappa$ est identique à $b\theta\kappa$, la clause $t\theta\neq b\theta$ peut s'unifier avex x=x pour produire la clause vide. Le système de règles reste donc complet si nous remplaçons la résolution par la:

Réfutation Réflexive:
Si dans la clause g≠d, g et d sont unifiables alors nous pouvons inférer la clause vide.

Introduisons également des règles de réduction dans ce cadre:
Simplification.
Voir Chapitre 4.
Subsomption Stricte.
Voir Chapitre 4.
Subsomption Fonctionnelle:
Si g[s]=g[t] et l=r sont deux équations telles qu'il existe une substitution σ vérifiant $l\sigma=g$ et $r\sigma=d$, alors g[s]=g[t] peut être supprimé.

Remarquons que la subsomption avec x=x correspond à la suppression des paires critiques triviales dans l'algorithme KB.

Nous appelerons *S-stratégie* l'ensemble des règles d'inférence: **Superposition Généralisée, Surréduction Généralisée, Réfutation Réflexive, Simplification, Subsomption Stricte, Subsomption Fonctionnelle.**

La complétude de la S-stratégie est un corollaire direct de la complétude des règles d'inférence de la section 1. Nous nous contentons de l'énoncer:

Théorème 1.1: Soit un ensemble E-inconsistant de clauses unitaires U ayant pour symbole de prédicat = et contenant la clause x=x. Toute dérivation équitable issue de U et obtenue par la S-stratégie contient la clause vide.

2. Une extension de l'algorithme de Knuth et Bendix.

Nous pouvons en particulier appliquer la stratégie précédente pour démontrer la validité d'une égalité s=t dans une théorie équationnelle A. En effet, soit $\sigma \neq \tau$ la clause obtenue en remplaçant chaque variable de s≠t par une nouvelle constante (skolémisation); alors:

$A \models s=t$ ssi $A \cup \{\sigma \neq \tau\}$ est E-inconsistant.

Supposons donc que la théorie U soit constituée uniquement d'un ensemble d'équations A et d'une clause sans variables $\sigma \neq \tau$. Appliquons la S-stratégie à U pour construire une réfutation. Montrons d'abord qu'il est toujours possible de contrôler l'application des règles d'inférences pour que le système ne contienne qu'une seule inégalité:

Supposons qu'il soit possible d'appliquer une surréduction généralisée à $\sigma \neq \tau$. Donc il existe un sous terme r de σ (ou τ) et une équation a=b vérifiant:
(1) $a\theta = r\theta$ avec θ unificateur principal de a et r,
(2) $a\theta \not\downarrow b\theta$,

Remarquons que aθ est un terme clos. En examinant la preuve de complétude de la stratégie, nous pouvons préciser les conditions (1) et (2) en:
(1'-2') il existe une substitution close π telle que $a\pi = r\pi$ et $a\pi > b\pi$.

Donc θ et π coincident sur les variables de a, et $a\theta=a\pi$. Prolongeons θ aux variables de b qui ne sont pas dans a, en leur associant la plus petite constante de l'univers de Herbrand. Par monotonie de l'ordre <, nous avons $b\pi > b\theta$ et donc aussi: $a\theta > b\theta$.

Par conséquent lorsqu'une surréduction généralisée est possible, une simplification est également possible. Nous choisirons d'appliquer cette dernière car elle réduit la taille du système à traiter, et permet de ne conserver qu'une seule inégalité.

Dans cette situation, les seules inférences possibles sont les suivantes:

Génération d'une équation.
1. Construire une paire critique généralisée entre deux équations, la simplifier le plus possible par les équations disponibles.
2. La supprimer si elle est subsumée par une autre équation, sinon (on dit que la paire critique *diverge*) essayer de l'orienter en règle de réécriture .

Réduction du but.

Simplifier $\sigma' \neq \tau'$ avec une équation:

1. si r est un sous-terme de s' unifiable avec le membre droit de l'équation a=b, d'unificateur principal θ, étendre θ aux variables de b qui n'appartiennent pas à a en leur associant la plus petite constante.

2. si $a\theta > b\theta$, remplacer $\sigma'[r] \neq \tau'$ par $\sigma'[b\theta] \neq \tau'$

Réfutation finale.

Lorsque $\varphi \neq \varphi$ est engendré, s'arrêter avec **succès**.

Nous appellerons procédure UKB tout algorithme appliquant de manière équitable les règles précédentes (aucune règle ne reste indéfiniment applicable sans être appliquée).

La procédure UKB est complète pour le problème du mot dans les théories équationnelles:

Théorème 2.1: Soit A une théorie équationnelle, s=t une équation valide dans A, et $\sigma \neq \tau$ la négation skolémisée de s=t. Alors UKB appliquée à A $\cup$ $\{\sigma \neq \tau\}$ s'arrête avec succès.

Nous allons maintenant retrouver les propriétés de l'algorithme de Knuth et Bendix. Admettons qu'il soit impossible de construire une paire critique généralisée non triviale à partir des équations de A. Supposons que s=t est valide dans A. Le système A $\cup$ $\{\sigma \neq \tau\}$ est E-inconsistant. Il en est de même pour le système A $\cup$ $\{\sigma' \neq \tau'\}$ où σ' et τ' sont obtenues en en normalisant σ et τ par A, c'est-à-dire en réduisant tant que c'est possible σ et τ par les équations de A. La seule règle d'inférence pouvant maintenant s'appliquer à A $\cup$ $\{\sigma' \neq \tau'\}$ est la règle de réfutation finale; autrement dit σ' et τ' sont identiques.

Définissons pour tout ensemble d'équations A une relation de réécriture $\rightarrow_A$ sur les termes clos comme la plus petite relation monotone contenant:

$$\{(l\sigma, r\sigma) ; (l,r) \in A, \sigma \text{ est une substitution close}, l\sigma > r\sigma\}$$

Les résultats précédents peuvent donc s'énoncer:

Théorème 2.2: Si aucune paire critique généralisée de A ne diverge, alors la relation de réécriture $\rightarrow_A$ est canonique sur les termes clos. Autrement dit, deux termes clos sont égaux dans la théorie A si leurs formes normales sont égales.

preuve: il suffit de remarquer que lorsque s=t est clos, sa négation skolémisée est $s \neq t$.

Exemple: soit la signature F={a,b,*} où a et b sont deux constantes et * un symbole d'arité 2. Soit la théorie A={x*y=y*x}. Puisque les paires critiques de A sont subsumées par la seule équation de A, $\rightarrow_A$ est canonique sur les termes clos. Si l'ordre de simplification utilisé < est le RPOS de status gauche-droite et a>b alors la forme normale de a*b est b*a.

Supposons maintenant que toutes les équations de A sont orientables en un système de règles R par >, et que A n' a pas de paires critiques (au sens classique) divergentes. Si l'équation s =t est valide dans A, alors en remplaçant les variables de s et t par de nouvelles constantes nous obtenons

les termes clos σ et τ. Les formes normales de σ et τ pour $\rightarrow_R$ sont identiques. Comme > est incrémental et σ et τ sont des instances de s et t, la normalisation de σ et τ peut se relever en une normalisation de s et t. Donc s et t ont également des formes normales identiques. Nous avons ainsi retrouvé le résultat classique:

Théorème 2.3 (Huet 1981): Si le système de réécriture R n'admet pas de paires critiques divergentes, alors il est canonique. Autrement dit, deux termes égaux (dans la théorie R) ont les mêmes formes normales.

2. Une implantation et quelques exemples.

Nous décrivons dans cette section une implantation de la S-stratégie et de la procédure UKB et nous montrons quelques exemples.

Cette implantation (appelée *Sbrêve*) a été réalisée par Jalel Mzali sur un Sun3/75 en langage CLU (Mzali 1986) à partir du laboratoire de réécriture *Rêve 2.4* (Forgard 1984).

Il contient toutes les règles d'inférence de cette section, sauf la subsomption, dont une version simplifiée a été implantée.

En général, les procédures de Knuth-Bendix traitent les équations ayant des variables différentes dans chaque membre par la technique du *splitting*. Supposons que les deux membres de l'équation l=r partagent les variables $x_1,\ldots,x_n$, et que l possède des variables qui ne sont pas dans r et réciproquement. Alors un nouveau symbole de fonction f est créé, ainsi que deux nouvelles règles $l \rightarrow f(x_1,\ldots,x_n)$ et $r \rightarrow f(x_1,\ldots,x_n)$. Cette technique est programmée dans Reve. Elle n'est pas nécessaire pour la procédure UKB, puisque cette dernière peut traiter les équations non orientables. Cependant le *splitting* peut réduire rapidement la complexité des termes en éliminant les variables non significatives.

Donnons d'abord un exemple d'application de la S-stratégie. Le problème est dû à Smullyan, selon Overbeek (Association of Automated Reasoning Newsletter 1985). Soit la théorie:

$S(x,S(y,x)) = S(f(x,y),z)$ e1

$S(m,x) = S(x,x)$ e2

$S(a,x) \neq x$ e3

Si l'ordre < est le RPOS à status gauche-droite, la première équation peut être orientée en règle:

$S(f(x,y),z) \rightarrow S(x,S(y,x))$ r1

La seule paire critique généralisée entre r1 et e2 s'obtient en superposant le membre gauche de r1 et S(x,x); en prenant m comme le plus petit symbole:

$S(x,S(y,f(x,y))) \rightarrow S(m,f(x,y))$ r4

Par surréduction généralisée entre r4 et e3, nous dérivons:

$S(m,f(x,y)) \neq S(y, f(x,y))$ e5

Les deux membres de e5 s'unifient (y/m). La règle de réfutation réflexive s'applique et permet de conclure à l'inconsistance.

Si e1 est ordonné dans l'autre sens (status gauche-droite), alors l'espace de recherche est plus grand car il y a plus de superpositions généralisées. Cependant la preuve est achevée après la génération de 6 équations, dont deux seulement sont utiles.

La preuve est obtenue par Sbreve en 4 secondes de CPU.

Donnons un exemple de complétion par la procédure UKB. Il s'agit de la théorie des

groupoides entropiques, complétée par une autre technique dans (Pedersen 1985).

Ces structures sont définies par les deux axiomes:

(x.y).(z.w) = (x.z).(y.w) e1

(x.y).x =x e2

Remarquons que le premier axiome est permutatif et ne peut pas être orienté. La procédure KB échoue sur ce problème, de même que les procédures de E-complétion (Jouannaud Kirchner 1986). La deuxième équation s'oriente:

(u.v).u →u r2

En unifiant u dans r2 avec (z.w) dans le membre gauche de e1, nous engendrons la paire critique:

((z.w).z).(y.w) = z.w

qui donne la règle:

z.(y.w) → z.w r3

r3 simplifie e1 en:

(x.y).w = (x.z).w e4

Le système suivant est obtenu finalement:

(u.v).u → u r2

z.(y.w) → z.w r3

(x.y).w = (x.z).w e4

((x.y).z).w → x.w r5

D'après le théorème ce système est canonique sur les termes clos. Sbreve a complété ce système en 6 secondes de CPU.Si nous utilisons la technique de splitting pour scinder l'équation e4 en deux règles, alors le programme diverge: il engendre une infinité d'équations. De nouveaux symboles de fonctions équivalents à des symboles déjà dans le système sont constamment introduits par splitting, indiquant que le processus boucle.

D'autres exemples sont donnés dans l'appendice de cette section.

3. Discussion.

Comparons notre approche avec celle de (Knuth et Bendix 1970) ou (Huet 1981) et leurs extensions.

1. Nous ne demandons pas d'avoir à chaque étape de l'algorithme UKB un système de règles orientées. Nous permettons même la présence d'équations ayant des variables différentes dans chaque membre. Si l'algorithme diverge, alors il fournit néanmoins une procédure de semi-décision pour le problème du mot, comme dans KB (Huet 1981).

Si le système final est fini alors il est canonique sur les termes clos; si de plus ses règles sont orientables, alors il est canonique.

2. Nous utilisons des ordres de simplification complets pour comparer les termes. La procédure KB utilise n'importe quel ordre de réduction bien fondé.

3. Pour traiter les équations non-orientables, les méthodes de E-complétion (Jouannaud Kirchner 1986) utilisent des algorithmes d'unification dans des théories contenant ces équations. Les procédures de E-complétion réussissent parfois à compléter des systèmes modulo une sous-théorie équationnelle. Cependant les algorithmes d'unification équationnelle sont rares et difficiles à trouver. Même lorsqu'aucun mécanisme spécial d'unification n'est connu, notre méthode peut obtenir des systèmes canoniques sur les termes clos (voir l'exemple des groupoides entropiques).

Cependant pour les théories associatives-commutatives (Peterson Stickel 1981)(Lankford Ballantyne 1977), les procédures de E-complétion s'arrêtent plus souvent avec succès que UKB. Nous pensons qu'une combinaison des deux méthodes (UKB + E-complétion) devrait allier leurs avantages respectifs.

4. Des résultats analogues à ceux de cette section ont été obtenus dans (Bachmair et al. 1986) par la technique des ordres bien-fondés sur les preuves. Etrangement, cette technique nécessite également des ordres de simplification totaux sur les termes clos, ainsi que des restrictions analogues aux nôtres sur la règle de simplification.

4. Appendice.

Les exemples suivants de systèmes canoniques comportant des règles non orientables ont été obtenus par Jalel Mzali sur Sbreve.

Exemple 1.

x*(x*x) = e

e*x = x

i(x) * x = e

x*y = y*x

h(x,y) = x * (y * (i(x) * i(y)))

UKB termine avec le système:

i(e) → e

e * x → x

x * e → y

x * i(x) → e

x*y = y*x

x*(x*x) → e

h(x,y) → x * (y * (i(x) * i(y)))

Exemple 2

(x * y) & (y * z) = (x * z) & (x * y)

x & y = y & x

précédence & > * et les deux symboles ont un status gauche-droite.

Le système termine avec:

x & y = y & x

(x * y) & (y * z) = (x * z) & (x * y)

(x * z) & (x * y) = (y * z) & (x * y)

(x * y) & (y * z) = (x * y) & (x * z)

(y * z) & (x * y) = (x * y) & (x * z)

(x * x) & (x * y) = (y * y) & (y * x)

(x * y) & (y * z) = (x * z) & (z * y)

$(z * y) \& (z * z) = (y * y) \& (y * z)$
$(x * y) \& (y * z) = (z * y) \& (x * z)$
$(y * z) \& (x * y) = (z * y) \& (x * z)$
$(x * y) \& (x * x) = (y * x) \& (y * y)$

Exemple 3.
$(a*a) + (a*a) = (b*a) + b$
$x* 1 = x$
$a*b = a+ 1$
$x*y = y*x$
$x+y = y+x$

Le système termine avec:
$x * 1 \rightarrow 1$
$1 * x \rightarrow 1$
$x*y = y*x$
$x+y = y+x$
$b*a \rightarrow a+1$
$(a*a) + (a*a) \rightarrow (a+1) + b$

SECTION 3: PROBLEMES DE MOTS DANS LES THEORIES DE HORN.

1. Introduction

La logique de Horn, qui est une restriction du calcul des prédicats du premier ordre, a fourni les bases théoriques de nombreuses applications en informatique: systèmes experts, spécifications algébriques, programmation logique, calcul formel ...Le langage Prolog a été developpé dans ce cadre puissant. Il y a eu plusieurs tentatives d'incorporer l'importante relation d'égalité dans Prolog: EQLOG (Goguen Meseguer 84) où des équations sont ajoutées aux Clauses de Horn, SLOG (Fribourg 85) qui autorise des clauses équationnelles. Toutes ces approches imposent de sérieuses restrictions aux programmes logiques qu'elles peuvent interpréter. Elles utilisent les équations comme des règles de réécriture pour simplifier les buts. Notre méthode est fondée la stratégie de réfutation complète de la section 1 de ce chapitre: une équation est considérée comme une règle de réécriture lorsqu'elle est orientable et quand ses pré-conditions lui sont inférieures, pour un ordre de simplification complet. Cette technique est plus souple que les approches classiques de réécriture conditionnelle, car

1. elle n'échoue pas lorsqu'apparait une règle non-orientable,
2. elle ignore les règles ayant des pré-conditions supérieures à leur conclusion.

De plus, un ensemble de clauses de Horn admet un modèle initial dans lequel un fait est vrai si et seulement si il peut être prouvé par les règles de déduction classiques de la logique du premier ordre. C'est le modèle le plus intéressant pour les bases de données (closed-world assumption (voir Reiter 78)), les types abstraits (algèbre initiale (Remy 82)(Padawitz 85)).

La règle de négation par échec (Clark 78) permet de raisonner dans ce modèle particulier. Nous proposerons une alternative à cette règle en étendant aux clauses de Horn la méthode de (Jouannaud Kounalis 86) pour démontrer des théorèmes inductifs dans les théories équationnelles.

2. Règles d'inférence pour les théories de Horn.

Nous avons démontré dans la section 2 que l'algorithme de Knuth et Bendix engendre toutes les conséquences d'un ensemble initial d'équations, au moyen d'un ensemble de règles d'inférence complet pour la logique équationnelle. La procédure de complétion peut donc s'interpréter comme une tentative de saturer un ensemble d'équations par un ensemble de règles d'inférence, en espérant obtenir un ensemble fini dont toutes les conséquences sont triviales (confluence locale).

L'intérêt de cette remarque est qu'elle peut s'appliquer aux clauses de Horn. Partant d'un ensemble de clauses de Horn et d'un ensembles de règles d'inférence, nous itérons ces règles sur les clauses. Si la procédure s'arrête, elle fournit un moyen très efficace de résoudre le problème du mot par normalisation conditionnelle, combinant à la fois le chaînage arrière (vérifier les pré-conditions avant d'appliquer une règle pour simplifier) et le chaînage avant (appliquer une règle pour simplifier).

Le littéral positif d'une clause de Horn est considéré comme une règle de simplification dont les pré-conditions sont rassemblées dans les littéraux négatifs.

E. Paul (1986) a construit un algorithme de saturation sur la base d'une stratégie unitaire. Mais des expériences montrent que sa procédure échoue même sur les ensembles les plus simples de clauses de Horn.

Nos résultats sont à rapprocher des techniques de réécriture conditionnelle (Remy 82) (Kaplan

87) (Jouannaud Waldmann 86) (Ganzinger 87) (Choquer 86). Mais à la différence de ces dernières, nous pouvons traiter des équations non-orientables ou des règles conditionnelles dont les pré-conditions sont plus complexes que la conclusion.

Les règles d'inférence que nous utiliserons dans cette section sont simplement des versions raffinées des règles de la section 1: la paramodulation et la superposition concerneront toujours les littéraux maximaux des clauses parentes. Pour simplifier nous condenserons ces deux règles en une seule appelée paramodulation.

O-Factorisation

O-Résolution

Paramodulation :

Soit $C_1 = (s=t) \vee D_1$ et $C_2 = C_2[n \leftarrow r]$, où r est un sous-terme non-variable à l'occurrence n du littéral L de C_2. Si

(1) $s\sigma = r\sigma$ avec σ unificateur principal de s et r,

(2) $s\sigma \not\leq t\sigma$,

(3) $(s=t)\sigma \not\leq A\sigma, \forall A \in D_1$

(4) $L\sigma \not\leq A\sigma, \forall A \in C_2 - \{L\}$,

(5) si L est une équation a=b et n est une occurrence de a, alors $a\sigma \not\leq b\sigma$

alors $C = C_2[n \leftarrow t]\sigma \vee D_1\sigma$ est un *paramodulant de C_1 dans C_2 à l'occurrence n.*

Remarque: l'application des règles précédentes à des clauses de Horn engendre toujours des clauses de Horn.

Remarque: considérons la clause $\neg$ s(a)=s(b) $\vee$ a=b. Il est interdit de l'utiliser comme une clause active dans une paramodulation, car son littéral positif est plus petit que sa partie conditionnelle. L'intérêt de cette restriction est de ramener le problème de la preuve d'une égalité à des sous-problèmes plus petits. Au contraire la règle de paramodulation règle favorise l'application de la clause $\neg$ a=b $\vee$ s(a)=s(b) car elle fait décroître la taille des problèmes.

Nous pouvons ajouter également les règles de réduction vues au Chapitre 4: **Subsomption Stricte, Simplification, Elimination des Tautologies, Simplification Clausale,** à condition d'imposer une hypothèse d'équité. Nous appellerons *RP* l'ensemble formé des règles de o-factorisation, o-résolution, paramodulation et *INF* l'ensemble obtenu en ajoutant à RP les règles de réduction précédentes.

Nous dirons qu'une clause est *normalisée* par l'ensemble de clauses S si elle est complètement simplifiée par les équations disponibles de S. Nous supposerons également qu'une clause est ajoutée au système courant uniquement si, après avoir été normalisée, elle n'est pas subsumée par une clause déjà présente dans ce système (*forward subsumption*).

Théorème de Complétude: toute dérivation équitable issue d'un ensemble E-inconsistant de clauses de Horn contenant x=x, contient la clause vide.

3. Ensembles de Horn saturés

Définition 3.1: ensemble saturé.
Un ensemble de clauses de Horn S est *saturé pour INF* si S est E-consistant et si aucune nouvelle clause ne peut être inférée à partir de S. Plus précisément, toute clause de RP(S), après normalisation, est une tautologie ou est subsumée par une clause de S.

La notion de saturation peut être considérée comme une généralisation de la notion de locale convergence des systèmes de réécriture. L'algorithme de saturation consiste à itérer l'application des règles d'inférences de INF à un ensemble initial S.

Exemple 1. L'ensemble de clauses suivant est saturé:

1. P(0) 2. $\neg$ P(x) $\vee$ P(s(x))

En effet nous avons vu dans la section 1.2 du Chapitre 3 qu'aucune résolution n'est possible sur ce système car P(x) < P(s(x)) pour tout ordre de simplification complet.

Exemple 2. Considérons maintenant une spécification du prédicat *inférieur ou égal* noté *i* dans l'ensemble des entiers (*s* et *p* sont les fonctions successeur et prédecesseur) (Kaplan 87):

0. x=x.
1. s(p(x)) = x.
2. p(s(x)) = x.
3. i(o,o) = t.
4. i(o,p(o)) = f.
5. $\neg$i(o,x) = t $\vee$ i(o,s(x)) = t.
6. $\neg$i(o,x) = f $\vee$ i(o,p(x)) = f.
7. i(s(x),y) = i(x, p(y)).
8. i(p(x),y) = i(x, s(y)).
9. $\neg$t=f.

L'algorithme de saturation appliqué à 0..9 avec le RPO lexicographique de (Dershowitz 87) et la précédence :i>p>s>o>t>f, s'arrête après avoir engendré deux nouvelles clauses:

10. $\neg$i(o,s(x)) = f or i(o,x) = f. par paramodulation de 2. et 6.
11. $\neg$i(o,p(x)) = t or i(o,x) = t. par paramodulation de 1. et 5.

Remarquons que les méthodes de réécriture conditionnelles sont incapables de traiter les règles 10. et 11. car leur pré-condition est plus grande que leur conclusion. Un autre aspect important de notre approche est l'utilisation de clauses négatives telles que 9. pour éviter la divergence du processus de saturation. Sans 9., une infinité de clauses serait engendrée; cependant, en présence de 9., ces clauses sont immédiatement subsumée. La clause $\neg$t=f est un théorème inductif de la spécification 1,...,8. L'adjonction de théorèmes inductifs à une spécification, peut être considérée comme une manière d'approcher le modèle initial par des formules du premier ordre. Cette idée a aussi été exploitée dans (Ganzinger 86).

Exemple 3. Tout système de réécriture convergent dont les règles sont orientées par un ordre de simplification complet.

D'autres exemples sont présentés en appendice.

Définition 3.2: Input dérivation.

Une *input-dérivation* D, d'un ensemble de clauses S, est une suite de clauses $(C_1,...,C_n)$ such that $C_1 \in S$ and each C_{i+1} est soit un facteur de C_i, soit un résolvant de C_i et d'une clause de S, soit un paramodulant d'une clause de S dans C_i. C_1 s'appelle la *clause-sommet* . Si C_n est la clause vide alors D est une *input-réfutation* de S.

Théorème 3.3: Soit C une clause négative et S un ensemble de Horn saturé (contenant x=x). Alors S ∪ {C} est E-inconsistant ssi il admet une input-réfutation de clause-sommet C.

preuve: la validité du système INF démontre la suffisance de la condition.

Supposons maintenant que S ∪ {C} est E-inconsistant, où S est un ensemble de Horn saturé (contenant x=x). Comme INF est complet, il existe une réfutation S ∪ {C}. La réfutation contient au moins une occurrence de C car S est E-consistant. Toute clause dérivant de C est nécessairement négative. Comme aucune inférence n'est possible entre des clauses négatives, C n'apparait qu'une fois dans la réfutation. Soit $(C_1,...,C_i,...,C_n)$ la suite des clauses dérivant de C dans la réfutation (C_1=C et C_n=[]). Si C_i n'est pas un facteur de C_{i-1}, alors un au moins des parents de C_i est une clause non-négative P_i; une clause de ce type ne peut s'obtenir qu'à partir de clauses de S (C n'a que des descendants négatifs). Cependant d'après l'hypothèse de saturation, aucune clause ne peut se déduire de S. Donc la clause P_i appartient à S, ce qui signifie que la réfutation est une input-réfutation.

4. Sémantique opérationnelle.

D'un point de vue formel, la version raffinée de paramodulation que nous utilisons s'interprète en terme de réécriture conditionnelle. Cependant, nous appliquons une clause à un terme clos comme une règle conditionnelle , uniquement si la clause admet une condition qui n'est pas plus grande que la conclusion.

Par exemple, les clauses 10. et 11. de l'exemple 2 ne seront jamais appliquées comme des règles de réécriture. Donc un ensemble saturé de clauses peut se décomposer en deux parties

1. La partie *statique* qui est l'union des clauses négatives et des clauses dont les conditions sont plus grandes que la conclusion. Aucune de ces clauses n'est jamais utilisée pour normaliser les termes.
2. La partie *opérationnelle* formée des clauses pouvant servir éventuellement de règles de réécriture.

Définitions 4.1:

Soit S un ensemble de Horn, nous définissons *STATIC(S)* comme l'ensemble suivant de clauses $\{\neg C \vee D \in S$ tel que pour chaque substitution close θ, $D\theta < C\theta\}$ et *OPER(S)* comme le complémentaire de STATIC(S) dans S. En particulier, STATIC(S) contient toutes les clauses négatives de S.

Définition 4.2: relation de réécriture conditionnelle.

Soit S un ensemble de Horn, A et B deux termes clos ou deux littéraux clos.

Soit $\neg C \vee s=t$ un élément de S, où C est une conjonction de littéraux positifs. Alors

$A \rightarrow B$ par $\neg C \vee s=t$ dans le contexte S

s'il existe une substitution θ telle que $A= A[s\theta]$, $B=A[t\theta]$, $s\theta > t\theta$

et une substitution close σ telle que
1. $S \models (C\theta)\sigma$,
2. $Dom(\sigma) \cap \{V(s) \cup V(t)\} = \emptyset$,
3. $(s\theta=t\theta) \notin C\theta$

Soit A un littéral et $\neg C \vee A'$ un élément de S, où C est une conjonction de littéraux positif et A' est un littéral positif. Alors

$A \rightarrow$ VRAI par $\neg C \vee A'$ dans le contexte S

s'il existe une substitution θ telle que $A = A'\theta$

er une substitution close σ telle que
1. $S \models (C\theta)\sigma$.
2. $Dom(\sigma) \cap V(A) = \emptyset$
3. $A'\theta \notin C\theta$

Nous écrivons *A -S->B* au lieu de A->B par une clause de S (dans le contexte de la théorie S). La clôture reflexive-transitive de *-S->* est notée *-S->**.

Remarquons que -S'-> et -S-> sont égales lorsque S' = OPER(S).

Définition 4.3:

Un ensemble de Horn *préserve les termes clos* si tout littéral équationnel positif s=t appartenant à une clause de S est orientable ou bien vérifie V(s)=V(t).

Dans la suite nous ne considèrerons que des ensembles de Horn qui préservent les termes clos.

Remarque 4.4:

La relation -S-> n'est pas décidable en général, car elle nécessite de vérifier la consistance d'une clause dans le contexte S.

Proposition 4.5:

Soit A un littéral positif clos et S un ensemble de Horn saturé (contenant x=x). Alors les deux propositions suivantes sont équivalentes:

1. $S \models A$
2. A -S->* VRAI

preuve: Supposons 1. Alors $\{\neg A\} \cup S$ est E-inconsistant. D'après le théorème 3.3, il existe une input-réfutation de $\{\neg A\} \cup S$ de clause-sommet $\neg A$. Raisonnons par induction noethérienne sur A, pour l'ordre de simplification <. Plusieurs cas sont à distinguer, selon la première règle d'inférence utilisée dans la réfutation.

cas 1: la première inférence est une résolution entre $\neg A$ et une clause $A' \vee D$ où D est une disjonction de littéraux négatifs et A' un littéral positif. Soit θ le filtre le plus général de A' vers A. Alors le reste de la réfutation est une réfutation of $D\theta$. Par conséquent, il existe une substitution σ telle que $S \models \neg D\theta\sigma$. Donc A -S->VRAI par $A' \vee D$.

cas 2: il existe un sous-terme sθ de A telle que la première inférence est une paramodulation entre ¬A[sθ] et la clause s=t V D, où θ est le filtre le plus général de sθ vers s. Comme A est clos, θ est close également; donc nous avons sθ>tθ. La clause obtenue après cette première inférence est: ¬A[tθ] V Dθ. Comme le reste de l'input- réfutation est une input-réfutation de cette clause, il existe une réfutation de ¬A[tθ] et une réfutation of Dθ. Nous en déduisons immédiatement:

(1) S |= A[tθ] (rappelons que A[tθ] est clos)

(2) S |= ¬Dθσ

pour une certaine substitution close σ.

Remarquons que A[tθ]<A[sθ] par monotonie de <. Donc, d'après l'hypothèse d'induction:

(3) A[tq] -S->* VRAI

La relation (2) valide la réécriture conditionnelle suivante:

(4) A[sθ] -> A[tθ] par s=t V D.

En rassemblant (3) et (4) , nous obtenons une dérivation de A vers VRAI.

Corollaire 4.6:

Si S |= s=t, où s et t sont clos, alors il existe un terme clos β tel que

s-S->* β *<-S-t.

C'est la *propriété de Church-Rosser* pour les ensembles de Horn saturés.

preuve: d'après la proposition, s=t -S->* VRAI. Nécessairement il existe une étape de réécriture à la racine du littéral s=t. Cette étape correspond à une résolution dans l' input-réfutation, et c'est la dernière étape. Donc, nous avons s=t -S->* a=b -S-> VRAI. Cependant une résolution sur un littéral d'égalité n'est autorisée que si l'autre clause est x=x. Cela implique que *a* est identique à *b*. Par conséquent nous pouvons prendre β égal à *a* .

Corollaire 4.7:

Si s *<-S- t -S->* u , où s,t et u sont clos, alors il existe un terme clos v tel que

s -S->* v *<-S- u.

C'est la propriété de *confluence sur les termes clos.*

preuve: si t -S->* s et t -S->* u alors S |= s=u par validité de la relation de réécriture. Nous pouvons alors conclure avec le corollaire précédent.

Corollaire 4.8:

Soit A_1 V A_2 V...V A_n une disjonction de littéraux positifs clos et S un ensemble de Horn saturé (contenant x=x). Alors les deux propositions suivantes sont équivalentes:

1. S |= A_1 V A_2 V...V A_n
2. A_1 V A_2 V...V A_n -S->* VRAI

preuve: une propriété connue des ensembles de Horn est que leurs conséquences logiques sont

toujours (équivalentes à) des clauses de Horn. Donc la condition 1. est équivalente à:

3. il existe i ∈ {1,...,n} tel que S |= Ai.

Le résultat s'en déduit par la proposition 4.5.

5. Problèmes de terminaison

Nous pourrons résoudre le problème du mot dans une théorie de Horn saturée par normalisation si nous éliminons les deux problèmes de terminaison suivants:

1. Terminaison de la réécriture.
2. Décidabilité de chaque étape de réécriture.

Le premier revient à montrer qu'il n'existe pas de chaîne infinie de réécriture par -S->. Avec nos hypothèses, nous évitons ce problème, car nous utilisons des ordres noethériens pour orienter les équations, à savoir les ordres de simplification, .

Pour appliquer une règle conditionnelle, il faut prouver les pré-conditions de la règle. Donc, nous avons besoin de prouver la terminaison des appels récursifs à notre démonstrateur.

Pour assurer l'arrêt de ce processus, nous ajoutons une nouvelle restriction sur l'ensemble S étudié, à savoir que les conditions d'une règle n'introduisent pas de nouvelles variables:

Définition 5.1: ensemble étroit

Un ensemble de clauses de Horn est étroit si pour toute clause C de S, nous avons

V(COND) ⊆ V(A)

où A est le littéral positif de la clause C et COND est la disjonction des littéraux négatifs de C.

Sous des hypothèses analogues, des systèmes de réécriture conditionnels ont été étudiés par (Remy 82)(Remy Zhang 84)(Kaplan 87) et (Jouannaud Waldmann 86).

L'importance des ensembles de Horn étroits apparait dans la proposition suivante:

Proposition 5.2:

La réductibilité d'un terme clos par -S-> est décidable quand S est un ensemble étroit saturé.

preuve: supposons que A soit égal à A'θ, où θ est le filtre le plus général de A vers A'; testons si A peut être réduit par ¬C V A'. Cela revient à prouver que S |= Cθ. Remarquons que Cθ est clos (d'après la condition 1.). Supposons, pour simplifier, que C ne contient qu'un littéral. D'après la proposition 4.5 , S |= Cθ est équivalent à Cθ -S->* VRAI.

Cependant, Cθ< A'θ, et donc Cθ<A. Par hypothèse d'induction (noethérienne), nous pouvons décider si Cθ -S-> * VRAI. La proposition s'en déduit.

Corollaire 5.3:

Sous les hypothèses de la proposition précédente, tout terme clos admet une unique forme normale pour la relation -S->.

Nous aurons besoin d'une hypothèse supplémentaire sur les ordres de simplification complets utilisables. Cette propriété a deja été mentionnée dans la section 2.2.4. du Chapitre 2:

Définition 5.4: Incrémentalité

Soit *GT(F,P)* la réunion des termes clos et des atomes clos construits sur les symboles de fonctions F et les symboles de relations P.

L'ordre de simplification complet <, défini sur *GT(F,P)*, est ***incrémental*** si, pour tout ensemble de nouveaux symboles de constantes F', il peut être prolongé en ordre de simplification complet sur *GT(F ∪F',P)*

La plupart des ordres utilisés en réécriture possède la propriété d'incrémentalité: RPOS, RDOS, PSO ... Nous supposerons dans la suite de ce travail que les ordres de simplifications complets ont cette propriété peu restrictive.

Proposition 5.5:

Le problème du mot est décidable dans les théories axiomatisées par des ensembles de Horn étroits et saturés qui préservent les termes clos.

preuve: soit S un ensemble de Horn étroit et saturé, et a et b deux termes.

Remarquons que S |= a=b ssi S |= aμ=bμ , où μ est une substitution qui remplace chaque variable de a=b par une nouvelle constante. Nous pouvons prolonger < à l'univers de Herbrand de {S,aμ=bμ} car < est incrémental. D'après le corollaire 5.3, il existe des termes clos irréductibles par -S-> , a' et b', tels que:

aμ-S->*a' et bμ-S->*b' .

Comme S est étroit, la relation de réécriture sur les termes clos est décidable; donc, les termes a'et b' sont accessibles. La proposition S |= aμ=bμ est équivalente à S |= a'=b'. La propriété de Church Rosser des ensembles saturés implique:

S |= a'=b' ssi a'=b' (syntaxiquement).

Quand le système S n'est pas étroit, nous pouvons tester les pré-conditions des règles conditionnelles par *narrowing* conditionnel comme dans (Kaplan 87) ou (Jouannaud Waldmann 86). Bien sûr, nous perdons alors la complétude.

Cependant, lorsque l'ordre de simplification est linéaire, (i.e. tout terme clos admet un nombre fini de minorants, comme la variante de l'ordre de Knuth et Bendix donnée dans (Peterson 83)), alors nous pouvons résoudre le problème du mot sans supposer que les ensembles saturés sont étroits. En effet il suffit d'essayer toutes les substitutions closes qui rendent ses pré-conditions plus petites que sa conclusion, pour savoir si une règle conditionnelle s'applique. Bien entendu, cette méthode énumérative est très inefficace.

6. Comparaison avec les autres approches.

A la différence des autres approches conditionnelles les pré-conditions de nos règles sont (équivalentes à) des conjonctions de littéraux positifs: les règles sont avant tout des clauses de Horn. Nous sommes donc assurés de l'existence d'un modèle. De plus il existe un modèle minimal analogue à l'algèbre initiale des théories équationnelles.

Le mécanisme de saturation est identique à la complétion conditionnelle. Cependant, notre procédure n'échoue pas en présence d'une équation non orientable ou lorsque le système n'est pas étroit. Ce qui importe, pour obtenir une procédure de décision, est de terminer sur un ensemble saturé, étroit, qui préserve les termes clos.

7. Preuves dans le modèle initial

Les ensembles de Horn admettent un modèle de Herbrand minimal, le modèle initial qui est le plus intéressant pour de nombreuses applications.

Définition 7.1:

Le *modèle initial* d'un ensemble de Horn est l'interprétation de Herbrand vérifiant pour tout atome clos A:

$$I(A) = VRAI \text{ ssi } S \models A$$

Une clause C est valide dans le modèle initial de S ssi toute instance close de C est un théorème de S. Il est en général impossible de prouver qu'une clause est valide (ou invalide) dans le modèle initial en utilisant seulement le système d'inférence INF: une forme de raisonnement par induction est également nécessaire.

Nous montrons dans cette section comment construire des preuves dans le modèle initial par une extension très simple de notre système d'inférence. Comme dans (Jouannaud Kounalis 86), nous exprimons la validité dans le modèle modèle initial de S par la relation de réduction induite par S. La clause C est valide dans le modèle initial de S ssi S et SU{C} ont le même modèle initial; mais ce modèle est caractérisé par les formes normales closes de -S->. Il nous suffit donc de montrer que la relation de réduction associée à SU{C} associe à chaque terme clos la même forme normale que par la relation de réduction -S->.

Avant d'étudier le cas général, nous considèrerons des ensembles de Horn sans égalité.

7.1. Ensembles de Horn sans égalité.

Définition 7.1.1:

Soit un ensemble de Horn S (sans égalité), une clause C est *inductivement réductible* par rapport à S ssi pour toute substitution close θ:

soit il existe un littéral positif P de C tel que Pθ est réductible par rapport à S,

soit il existe un littéral négatif ¬N de C tel quet Nθ est irréducible par rapport à S.

Exemple 7.1.2:

Soit S l' ensemble{P(0), ¬P(x) ∨ P(s(x))}. Alors l'atome P(x) est inductivement réductible par rapport à S.

Le théorème clef de cette méthode est:

Théorème 7.1.3:

Soit S un ensemble de Horn saturé (sans égalité) qui préserve les termes clos. Alors une clause C est valide dans le modèle initial de S ssi C est inductivement réductible par rapport à S.

Exemple 7.1.4:

Soit S l'ensemble saturé { Eq(x,x), i(0,s(x)), ¬i(x,y) ∨ i(s(x),s(y))}

où *i* représente la relation *inférieur* sur les entiers naturels. Les formules suivantes sont valides dans le modèle initial de S, car elles sont inductivement réductibles:

i(x,s(x)) , ¬i(x,x) , i(x,y) V i(y,x) V Eq(x,y) (loi de trichotomie).
Noter que la dernière formule n'est pas une clause de Horn.

7.2. Ensembles de Horn avec égalité.

Etendons la méthode précédente afin de pouvoir traiter l'égalité. Rappelons que les formes normales closes pour la relation -S-> ne sont pas modifiées lorsque S est augmenté d'une clause valide dans son modèle initial.

Définition 7.2.1:
Soit un ensemble de Horn S. Un littéral A positif non équationnel (resp. une équation s = t avec s > t) est *inductivement réductible* par rapport à S ssi toutes les instances closes de A (resp de s) sont réductibles par rapport à S.

Lemme 7.2.2:
Soit S un ensemble de Horn qui préserve les termes clos et A un littéral positif qui est inductivement réductible par rapport à S. Alors un terme clos ou un atome clos est en forme normale pour S ssi il est en forme normale pour S U {A}.

Si nous considérons seulement des ensembles saturés, les propriétés de confluence de la section 4 montrent que l'adjonction d'un atome inductivement réductible ne modifie pas le modèle initial:

Théorème 7.2.3:
Soit S ensemble de Horn saturé qui préserve les termes clos et A un littéral positif. Supposons que S U {A} est saturé aussi. Alors A est valide dans le modèle initial de S ssi A est inductivement réductible par rapport à S.

7.3. Vérification de la réductibilité inductive.

La réductibilité inductive est en général indécidable. Pour quelques classes intéressantes d'ensembles de Horn (sans égalité) nous avons construit un algorithme de décision fondé sur la notion d'ensemble test, telle qu'elle est décrite dans le cas équationnel (Jouannaud Kounalis 86): vérifier la réductibilité inductive d'un objet revient à tester la réductibilité d'un nombre fini d'instances closes de cet objet. Les instances closes à examiner ont une profondeur déterminée par la profondeur des littéraux de S. Voir (Kounalis Rusinowitch 1987).

Exemple 7.3.1:
Soit S l'ensemble{ P(0) , ¬P(x) V P(s(x)) }. Pour l'atome P(x) , nous pouvons prendre pour ensemble test {P(0), P(s(0)). P(s(s(0))), P(s(s(s(0))))}. Chaque atome de l'ensemble test est réductible et donc P(x) est inductivement réductible par rapport à S.

8.Appendice

8.1. Voici une spécification des listes ordonnées. Les flèches indiquent les clauses opérationnelles; ces clauses sont les seules nécessaires pour normaliser les termes clos; elles forment un système convergent sur les termes clos:

0. x = x.
1. ¬t=f.
2. ¬i(x,y)=t V ¬i(x,y)=f.
-> 3. ¬i(x,y)=f V i(y,x)=t.
4. ¬i(x,y)=f V ¬i(y,x)=f .
5. ¬i(x,x)=f.

-> 6. ¬i(x,y) =t V max(x,y)=y.
-> 7. ¬i(x,y) =f V max(x,y)=x.
-> 8. i(x,max(x,y))=t.
-> 9. i(y,max(x,y))=t.

-> 10. ordered(nil).
-> 11. ordered(cons(x,nil)).
-> 12. ¬i(x,y)=t V ¬ordered(cons(y,z)) V ordered(cons(x,cons(y,z))).
13. ¬i(x,y)=f V ¬ordered(cons(x,cons(y,z))).

-> 14. insert(x,nil)=x.
-> 15. ¬i(x,y)=t V insert(x,cons(y,z))=cons(x,cons(y,z)).
-> 16. ¬i(x,y)=f V insert(x,cons(y,z))= cons(y,insert(x,z)).

Nous avons démontré la convergence de cet ensemble de clauses en orientant les termes par le RPOS de status gauche-droite et la précédence:
insert> cons > max > i > t >f >nil.

8.2. Voici une spécification des fonctions *pair* et *impair* sur les entiers:
0. x=x
1. even(o)=t.
2. even(s(o))=f.
3. even(s(s(x)))=even(x).
4. ¬even(x)=t V odd(x)=f.
5. ¬even(x)=f V odd(x)=t.
6. ¬t=f.

Si nous essayons de saturer cet ensemble de clauses, par le RPOS et la précédence:
odd > even > s > o >t >f, nous obtenons la clause

7. ¬even(x)=t or ¬even(x)=f or t=f.

Cette clause est simplifiée (simplification clausale) par 6. en:

7'. ¬even(x)=t or ¬even(x)=f

Les clauses 1..5 forment un système convergent sur les termes clos.

CHAPITRE 6

Ensembles complets de règles d'inférence pour les axiomes de régularité.

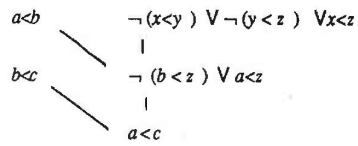
1. Introduction.

L'introduction de la paramodulation pour éviter d'appliquer la résolution avec les axiomes d'égalité peut se généraliser. Il est en effet possible d'incorporer d'autres théories que l'égalité dans des mécanismes d'inférences ad hoc, de sorte que leurs axiomes ne soient jamais requis dans une déduction. L'espace de recherche et la longueur des preuves en sont considérablement réduits. La procédure de preuve peut simuler ainsi un raisonnement à deux niveaux:

- un niveau spécialisé relatif à la théorie construite dans le mécanisme d'inférence, par exemple le raisonnement équationnel,
- un niveau universel pris en compte par la règle de résolution.

Stickel (1985) a développé une étude systématique de cette approche, appelée "*theory resolution.*".

De nombreuses théories axiomatiques appellent des schémas de raisonnement d'usage fréquent. Par exemple en théorie des ordres, on infére a<c de a<b et b<c par transitivité. Si l'on exprime ce raisonnement par la seule règle de résolution, on obtient:



Remarquons également que de a<b et de l'axiome de transitivité, on peut engendrer une infinité de conséquences par résolution. Il est plus naturel et plus efficace de construire une règle du type:

$$\frac{u<v \ \mathrm{V}\, R \qquad v<w \ \mathrm{V}\, Q}{u<w \ \mathrm{V}\, R \ \mathrm{V}\, Q}$$

et d'exclure la transitivité de l'ensemble initial des clauses. Cette fois, on ne peut engendrer qu'une seule conséquence et la déduction ne comporte qu'une seule étape.

Des règles d'inférence complètes sont connues pour les ordres et la théorie élémentaire des ensembles (Slagle 1972) (Slagle Norton 1975), pour les ordres totaux denses (Bledsoe Hines 1980) (Bledsoe et al. 1985), pour les relations binaires spéciales (Manna Waldinger 1986).

Plotkin (1972) a montré comment incorporer des axiomes équationnels, comme la commutativité ou l'associativité dans le processus d'unification, qui est à la base de la résolution, pour obtenir des procédures complètes. Slagle à proposé de construire les théories équationnelles dans le processus de déduction par la règle de surréduction ("narrowing"), qui n'est autre que la paramodulation orientée suivie de normalisation. Comme le remarque Fages (1983), la procédure de Slagle est une instance de celle de Plotkin, avec pour algorithme d'unification équationnelle celui de Fay (1979), qui procède justement par surréduction.

Nous allons considérer maintenant le point de vue sémantique, en cherchant à construire les modèles adaptés aux théories que l'on souhaite incorporer dans les règles d'inférence. Nous avons remarqué, en logique du premier ordre avec égalité, qu'il suffisait de raisonner sur les modèles de Herbrand égalitaires ou E-interprétations. En effet un ensemble de clauses est consistant avec la théorie de l'égalité si et seulement si il est validé par une E-interprétation (cf. Chang Lee 1973).

Considérons plus généralement un ensemble d'axiomes C. Définissons une C-interprétation comme un modèle de Herbrand qui valide C. Par exemple, pour C = {x.y=x.z ⇒ y=z} ∪ E, une C-interprétation est une E-interprétation I qui vérifie de plus I(a.b=a.c) = I(b=c) quels que soit les termes clos a,b et c.

Comme pour l'égalité le théorème de Herbrand s'adapte immédiatement: un ensemble de clauses est consistant avec la théorie C si et seulement si il est validé par une C-interprétation . Pour obtenir une construction incrémentale des C-interprétations, nous supposerons l'existence d'un ordre sur l'univers de Herbrand, tel que que la valeur d'un atome pour une C-interprétation ne dépende que des valeurs des atomes le précédant.

Si nous reprenons l'exemple précédent, nous remarquons que les C-interprétations incrémentales ne peuvent se définir simplement comme des ET-interprétations (cf. chapitre 2):
Supposons qu'une interprétation **I** soit définie pour tout atome strictement inférieur à (a.b=a), et que b>a (pour un ordre CSO). Supposons également que I(b=a)=**F** , I(a.a=a)=**V** et que (a.b=a) ne soit pas réductible par une équation valide de I. Dans la construction classique des E-interprétations, la valeur de I(a.b=a) peut être prise aussi bien égale à **V** ou **F**. Cependant si I(a.b=a) vaut **V**, il sera impossible de prolonger I de manière consistante avec C : par les axiomes d'égalité I(a.b=a.a)=**V**, et par l'axiome de régularité I(b=a)=**V**. Par conséquent dans la situation décrite par les hypothèses, il est nécessaire de donner à (a.b=a) la valeur **F**.

L'analyse de la construction des C-interprétations nous a permis de découvrir des règles d'inférence suffisantes pour circonvenir *complètement* l'axiome de régularité. De nombreux autres axiomes autorisent des constructions semblables.

2. Règles d'inférence pour les axiomes de régularité.

Nous présentons trois types de règles de régularité: régularité simple, régularité avec identité, régularité sauf pour l'élément nul.

2.1. Régularité à droite.

Une fonction f est régulière à droite si elle vérifie la loi de régularité à droite:

$$\forall x,y,z \ (f(y,x)=f(z,x) \supset y=z \ ).$$

Pour simplifier nous ne considèrerons que ce type de régularité. La régularité à gauche peut être traitée de manière symétrique. Cet axiome, s'il est juste ajouté à l'ensemble de clauses donné, peut engendrer de nombreuses nouvelles clauses par résolution et paramodulation. En le remplaçant par des règles d'inférence, nous n'engendrerons que les clauses les plus significatives. Dans la suite nous donnons une version peu raffinée des règles d'inférence. Une version optimisée, introduisant un ordre CSO sur les termes sera donnée plus loin. Deux listes $(s_1,\ldots,s_n)$ et $(t_1,\ldots,t_n)$ sont unifiables avec pour unificateur principal σ, ce qui s'écrit: $(s_1,\ldots,s_n)\sigma = (t_1,\ldots,t_n)\sigma$, si σ est l'unificateur le plus général vérifiant $s_i\sigma=t_i\sigma$ quel que soit i .

$$\text{C1}\quad \frac{(s=t\ ) \vee C}{(y=z\ )\sigma \vee C\sigma} \qquad \text{où } (s,t)\sigma=(f(y,x), f(z,x)\ )\sigma$$

$$\text{C2}\quad \frac{(s=t\ ) \vee C,\ (l=r\ ) \vee D}{(y=z\ )\sigma \vee C\sigma \vee D\sigma} \qquad \text{où } (s,t,l,r)\sigma= (f(y,x),\ u,\ f(z,x),\ u\ )\sigma$$

La validité des règles d'inférences C1 et C2 est facile à établir. Ces règles, avec la résolution, la factorisation et la paramodulation forment une stratégie complète pour le calcul des prédicats du premier ordre avec égalité et avec des symboles réguliers à droite . Le seul axiome requis est $x=x$.

Montrons l'utilisation des axiomes sur un exemple très simple. Soit les deux clauses:

$(b+y\ )+a=y \quad c1$

et

$b+(a+a\ )\neq a \quad c2$

avec + régulier à droite. La seule règle applicable est C1 sur la clause c1, produisant:

$b+(x+a\ )=x. \quad c3$

Par résolution de c3 et c2, on obtient une contradiction .

2.2. Régularité avec Identité.

Des règles d'inférence analogue s'obtiennent pour d'autres axiomes de régularité. Par exemple, il est fréquent de rencontrer en algèbre un élément *identité e* pour un opérateur, satisfaisant la loi de régularité suivante:

$$\forall xy \ (f(y,x)=x \supset y = e\ )$$

Bien que cette loi résulte de la régularité à droite et de l'axiome d'identité à gauche, il peut être intéressant de l'étudier indépendamment, comme un axiome, sans les deux autres. L'opérateur + dans la théorie des anneaux satisfait cet axiome, et Stickel a montré le gain d'efficacité considérable

lorsqu'il est exprimé par des règles d'inférence(Stickel 1984).

La règle C1 peut être modifiée en:

$$\text{CI1}\quad \frac{(s=t)\vee C}{\sigma(y)=e\vee C\sigma}$$

$$où\ \ (s,t)\sigma=(f(y,x),x)\sigma$$

Cette seule règle, et, bien sûr la résolution, la factorisation et la paramodulation forment une stratégie complète pour le calcul des prédicats avec égalité et certains opérateurs réguliers à droite avec identité. Remarquons qu'un axiome du type de CI1 est introduit pour chacun de ces opérateurs.

La règle C2 peut aussi être modifiée en:

$$\text{CI2}\quad \frac{(s=t)\vee C,\ (l=r)\vee D}{(y=e)\sigma\vee C\sigma\vee D\sigma}$$

$$où\ \ (s,t,l,r)\sigma=(f(y,x),u,x,u)\sigma$$

Cependant elle n'est pas nécessaire à la complétude.

2.3. Régularité sauf l'élément nul.

Dans de nombreuses théories (domaines d'intégrité, corps) la loi de régularité d'un opérateur est vérifiée sauf pour un élément nul.

Soit f un opérateur et 0 son élement nul (autrement dit, $f(0,x)=0$ pour tout x), la loi de *régularité sauf l'élément nul* est:

$$\forall x\,y\,z\ \ (x\neq 0\ \wedge f(y,x)=f(z,x)\supset y=z)$$

En modifiant les règles d'inférences de la régularité à droite pour tenir compte de la condition $x\neq 0$ nous obtenons les règles d'inférence suivantes:

$$\text{CN1}\quad \frac{(s=t)\vee C}{(y=z)\sigma\vee C\sigma\vee(x=0)\sigma}$$

$$où\ \ (s,t)\sigma=(f(y,x)f(z,x))\sigma$$

$$\text{CN2}\quad \frac{(s=t)\vee C,\ (l=r)\vee D}{(y=z)\sigma\vee C\sigma\vee(x=0)\sigma}$$

$$où\ \ (s,t,l,r)\sigma=(f(y,x),u,f(z,x),u)\sigma$$

Comme exemple simple, nous montrons qu'un anneau commutatif est un domaine intègre si * est régulier à droite sauf l'élément nul. En d'autres termes, nous voulons prouver que:

$$\forall x\,y\ \ (x*y=0\supset x=0\vee y=0)$$

Après skolémisation du but, les données contiennent:

$0*x=0$	$c1$
$a*b=0$	$c2$
$a\neq 0$	$c3$
$b\neq 0$	$c4$

En appliquant $CN2$ sur $c1$ et $c2$, nous obtenons

$$a=0\vee b=0, \qquad c5$$

ce qui, avec $c3$ et $c4$, conduit immédiatement à une contradiction.

3. Règles d'Inférences avec Ordre de Simplification Complet.

Les règles d'inférence décrites plus haut peuvent être améliorées considérablement (tout en conservant la complétude) si l'on introduit un ordre CSO sur les termes. De plus, l'utilisation des règles de O-paramodulation et démodulation contribuent encore à l'accroissement de l'efficacité.

Les CSO permettent de comparer les littéraux dans une clause et les membres d'une équation. Ils permettent de limiter les inférences aux éléments maximaux des données. Au Chapitre 3 la paramodulation, la résolution et la factorisation ont été raffinés grâce aux CSOs. On peut restreindre de la même manière les règles d'inférence pour les axiomes de régularité. Les règles d'inférence $C1$ et $C2$ deviennent:

$$\text{SC1}\quad \frac{(s=t)\vee C}{(y=z)\sigma \vee C\sigma}$$

où $(s,t)\sigma = (f(y,x),f(z,x))\sigma$, $(s=t)\sigma \notin L\sigma \ \forall L \in C$

$$\text{SC2}\quad \frac{(s_1=t_1)\vee C_1, (s_2=t_2)\vee C_2}{(y=z)\sigma \vee C_1\sigma \vee C_2\sigma}$$

où $(s_1,t_1,s_2,t_2)\sigma = (f(y,x),u f(z,x),u)\sigma$, et $(s_i=t_i)\sigma \notin L\sigma \ \forall L \in C_i$

Il faut noter que si le CSO utilisé compare les prédicats d'abord (= étant le plus petit de ces prédicats) aucune inférence de type SC1 ou SC2 ne sera appliquée à une clause contenant d'autres prédicats que l'égalité (car tout littéral égalitaire est plus petit que tout littéral sans égalité, voir Chapitre2).

Les règles ci-dessus, plus les raffinements de la résolution, factorisation et paramodulation construits sur l'ordre forment une stratégie complète:

Théorème 3.1: Soit un ensemble de clauses S contenant $x=x$, S n'est valide dans aucun modèle de la théorie formée des axiomes d'égalité et de l'axiome de régularité à droite ssi on peut engendrer *NIL* à partir de S avec les règles *SC* 1-*SC* 2, *O-résolution, O-paramodulation, O-factorisation.*

La preuve nécessite d'introduire des arbres transfinis représentant l'ensemble de tous les modèles canoniques de l'axiome de régularité (et de la théorie égalitaire). Remarquons que ni les axiomes réflexifs fonctionnels ni l'axiome de régularité ne sont requis.

Les autres règles d'inférence pour la régularité peuvent aussi être modifiée pour tenir compte du CSO.

Dans ce chapitre nous avons présenté plusieurs ensembles complets de règles d'inférence qui remplace les axiomes de régularité. L'avantage de ces règles spécialisées est qu'elles engendrent moins de conséquences redondantes et conduisent à des preuves plus directes. Nous avons amélioré ces règles en utilisant un CSO. Il est également possible d' ajouter des règles de réduction, comme au Chapitre 4, sans perdre la complétude. Les règles d'inférence de la régularité se sont aussi révélées précieuses pour accélérer la convergence de la procédure de Knuth-Bendix. L'implantation d'un prototype a permis d'obtenir un système canonique pour les groupes après génération de 14 équations au lieu des 17 habituelles. (Le système final contient 10 règles comme dans (Knuth Bendix 1970)). La cause principale de cette amélioration est la dérivation immédiate de la règle $(-x)+(x+y)\rightarrow y$, qui se trouve dans le système final, à partir de l'équation $x+((-x)+y)\rightarrow y$ et d'une règle de

distributivité gauche.

Dans (Knuth Bendix 1970) les axiomes de régularité simple sont construits dans un cadre équationnel, en ajoutant un nouveau symbole de fonction pour chaque opérateur régulier. Cette méthode s'étend facilement au premier ordre. Comme cette approche permet de simuler nos règles d'inférence, elle est également complète, mais il semble qu'elle engendre plus de clauses.

Une règle d'inférence appelée *paramodulation négative* pour traiter la régularité à été introduite par (Wos MacCune 1986). Cette règle ne permet pas d'éliminer complètement les axiomes de régularité. Un simple contre-exemple est l'ensemble inconsistant $\{a+c=b+c, h(a)\neq h(b)\}$,où + est régulier à droite. Ni la paramodulation négative, ni la paramodulation, ni la résolution ne sont applicables à aucune clause. Cependant la paramodulation négative est compatible avec les règles d'inférence pour la régularité que nous avons introduites. Cette règle devrait améliorer les performances de nos stratégies car elle introduit une certaine forme de recherche en chainage arrière.

4. Preuve de complétude des règles d'inférence pour les axiomes de régularité.

4.1. Régularité à droite.

Soit f un symbole de fonction qui vérifie l' axiome:

$$(C) \quad \forall x, y, z \ f(y,x)=f(z,x) \Rightarrow y=z$$

Pour simplifier les notations, nous omettrons le symbole f et nous écrirons ab au lieu de $f(a,b)$.

Théorème 4.1: Un ensemble de clauses S est inconsistant par rapport aux axiomes d'égalité et de régularité à droite si et seulement si une contradiction peut être obtenue de $S \cup \{x=x\}$ avec les règles d'inférence *SC*1, *SC*2, *O-résolution, O-paramodulation et O-factorisation.*

La preuve suit le schéma habituel. L'ensemble des règles d'inférence sera noté *CRP*.

4.1.1. Construction de Modèles Canoniques pour l'Axiome de Régularité à Droite.

Une interprétation de Herbrand I est une *C-interprétation* si c'est une E-interprétation telle que:

$$\forall\, a, b, c \in T(F),\ I(ac=bc)=I(a=b).$$

Notons qu'une *C-interprétation* n'est autre qu'une interprétation de Herbrand qui est un modèle de $K \cup \{C\}$ où K est l'ensemble des axiomes d'égalité. Nous pouvons établir une version du Théorème de Herbrand pour la théorie $K \cup \{C\}$:

Théorème 4.2: Soit S un ensemble de clauses. Alors $S \cup K \cup \{C\}$ est inconsistant ssi S n'est valide dans aucune *C*-interprétation.

preuve: voir (Chang Lee 1973).

Nous supposons donné un ordre complet de simplification $<$ sur $T(F,X) \cup A(P,F,X)$. Cela nous permet de construire les C-interprétations par induction, par rapport à cet ordre bien fondé:

Définition 4.3: Une interprétation I est une *CT*-interprétation si c'est une *E-interprétation* et si pour tous les termes t, a, b, c de $T(F)$ vérifiant les 4 conditions suivantes:

1. $ac=t$ est *I-irréductible*

2. $t < ac$

3. $I(t=bc)=\mathbf{V}$

4. $bc \leq ac$

nous avons également $I(ac=t)=\mathbf{F}$

Lemme 4.4: I est une *C-interprétation* ssi I est une *CT*-interprétation.

$\Rightarrow$ Supposons que $ac=t$ est *I-irréductible*, $t < ac$, $I(t=bc)=\mathbf{T}$, $bc \leq ac$.

Supposons que $I(ac=t)=\mathbf{T}$, comme $I(bc=t)=\mathbf{T}$ alors $I(ac=bc)=I(t=t)=\mathbf{T}$. D'où $I(a=b)=\mathbf{T}$, car I est une *C*-interprétation. Mais cela contredit l'hypothèse selon laquelle $ac=t$ est *I-irréductible.*

$\Leftarrow$ Soit I une *CT*-interprétation. Montrons que

$$\forall\, a, b, c \quad I(ac=bc)=I(a=b). \quad (*)$$

Supposons que cela soit faux. Soit alors $ac=bc$ le plus petit atome tel que (*) n'est pas vérifié. Plusieurs cas sont à considérer: (nous supposons que $ac > bc$)

cas 1: $I(ac=bc)=\mathbf{F}$ et $I(a=b)=\mathbf{V}$. Ce cas ne peut avoir lieu car I est une *E-interprétation.*

cas 2: $I(ac=bc)=\mathbf{V}$ et $I(a=b)=\mathbf{F}$.

Par définition d'une *CT*-interprétation, $ac=bc$ est *I*-réductible.

cas 2.1: $a=b$ est *I-réductible.*

Notons que le terme a peut être *I* -réduit en un terme a'. Nous avons

$$(a'c=bc) < (ac=bc),\ I(a'c=bc)=\mathbf{V} \text{ et } I(a=b)=\mathbf{F}.$$

Cela contredit l'hypothèse selon laquelle $ac=bc$ est le plus petit contre-exemple de (*).

cas 2.2: c est *I*-réductible: comme en 2.2 nous pouvons construire un contre-exemple plus petit.

cas 2.3: a, b, c sont *I*-irréductible et $ac=bc$ est *I*-réductible.

cas 2.3.1: il existe $ad < bc$ tel que $I(ac=d)=\mathbf{V}$.

Prenons pour d le plus petit terme vérifiant les hypothèses.

De $I(ac=bc)=\mathbf{V}$ nous déduisons $I(bc=d)=\mathbf{V}$ et $ac=d$ est *I*-irréductible.

Ceci est impossible car la définition d'une CT-interprétation implique $I(ac=d)=\mathbf{F}$.

cas 2.3.2.: il n'existe pas de d, tel que $d < bc$ et $I(ac = d) = \mathbf{V}$.

Comme $ac=bc$ est *I-réductible*, il existe un terme d' ($<bc$) avec $I(bc=d')=\mathbf{V}$. Prenons d' le plus petit possible. Alors $bc=d'$ est *I*-irréductible. Donc il en est de même pour $ac=d'$, ce qui permet de conclure comme dans le cas précédent.

Définition 4.5: arbre *C*-sémantique transfini.

L'arbre *C-sémantique transfini* est l'ensemble de toutes les *C-interprétations* que nous pouvons définir sur l'univers de Herbrand, ordonné par l'ordre de prolongement des applications.

Soit un arbre *C*-sémantique transfini *CT*, nous appelons arbre *C*-sémantique consistant maximal

d'un ensemble de clauses S, noté $MCCT(S)$, le sous-arbre maximal de CT tel que pour tout noeud I de MCCT(S), toute clause C de S et toute substitution close θ tels que les atomes de $C\theta$ sont dans le domaine de I, $I(C\theta)$=T.

4.1.2. Preuve dans le cas des termes clos.

Supposons que CRP n'est pas une stratégie complète, alors il existe un ensemble de clauses S qui n'est valide dans aucune *C-interprétation* et tel que MCCT(S *) est vide. Considérons la branche droite de MCCT(S*) et I le dernier noeud de cette branche. Chaque successeur immédiat de I est un noeud d'échec. Soit B un atome qui est la borne supérieure du domaine de I.

cas 1. si I a deux successeurs immédiats, alors par O -résolution sur les clauses qui étiquettent ces noeuds d'échec, il est possible de produire une clause de S * falsifiée par un ancêtre de I.

cas 2. si I a seulement un successeur et B est *I-réductible*, alors une étape de O-paramodulation engendre une clause de S * falsifiée par un ancêtre de I.

cas 3. supposons que I admet un seul successeur J et B est *I-irréductible*.

Nécessairement, il existe des termes a, b, c, t tels que:

$$B \equiv (ac=t),\ bc \leq ac,\ I(ac=t) = \mathbf{F} \text{ et } I(bc=t) = \mathbf{V}.$$

Supposons quebc est le plus petit terme vérifiant l'hypothèse précédente.

cas 3.1: $t \equiv bc$

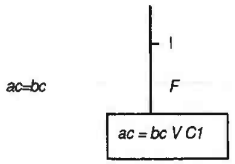


figure 1

Le noeud d'échec I peut être étiqueté par une clause $ac=bc \vee C1$ de S *. Notons que $ac=bc > C1$. Si nous appliquons la règle $SC1$, nous obtenons la clause $a=b \vee C1$.

Remarquons que $I(C1)$=F. Comme $ac=bc$ est I-irréductible, alors $I(a=b)$=F. Donc $a=b \vee C1$ es falsifiée par un ancêtre de I.

cas 3.2: bc < t

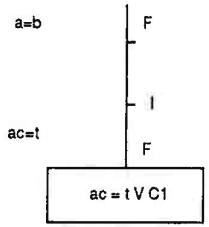


figure 2

Le noeud d'échec J peut être étiqueté par une clause $ac=t \vee C1$ de S *. Comme $ac=t$ est I-irréductible, nous avons $I(a=b)$=F.

Lemme 4.6: il n'existe pas de terme a' tel que $a' < a$ (resp. $b' < b$, resp. $c' < c$) et $I(a=a')$=V (resp. $I(b=b')$=V, resp. $I(c=c')$=V).

preuve: si $I(a=a')$=V ou $I(c=c')$=V, $ac=t$ serait *I-réductible*. Si $I(b=b')$=V, $b'c$ serait un terme plus petit que bc vérifiant les hypothèses.

Soit N la restriction de I à W($bc=t$).

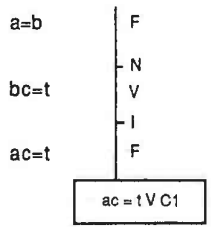


figure 3

Lemma 4.7: $bc=t$ est *N-irréductible*.

preuve: Supposons qu'il existe un terme g tel que $g < t$ et $N(bc=g)$=V. Alors

$$N(bc=t) = N(bc=g) = N(t=g) = \mathbf{V}.$$

Cela implique que $ac=t$ est I-réductible. La même conclusion reste valide si l'on suppose qu'il existe un g tel que $g < t$ et $N(t=g)$=V.

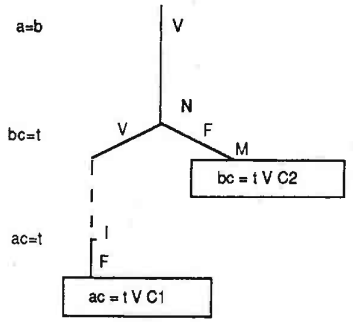


figure 4

Comme $bc=t$ est N-irréductible, N admet deux successeurs. Le successeur droit M est forcément un noeud d'échec car la branche droite mène à I. Donc M peut être étiqueté par une clause $bc=t \vee C2$. Par application de la règle $SC2$ à $bc=t \vee C2$ et $ac=t \vee C1$, nous obtenons $a=b \vee C1 \vee C2$. Il est facile de vérifier que $I(a=b \vee C1 \vee C2)=\mathbf{F}$. Un ancêtre de I falsifie donc une clause de S*, ce qui est contradictoire.

4.1.3. Relèvement des Inférences.

Nous devons montrer que chaque clause inférée à partir d'instances closes de clauses est une instance d'une clause obtenue au niveau général.

Lemme de Relèvement de SC1: Soit C la clause $g=d \vee C'$. Supposons que (g, d) et (xz, yz) ont pour unificateur principal σ. Supposons que θ est une substitution telle que $g\theta = ac$, $d\theta = bc$.

Alors la clause $a=b \vee C'\theta$ est une instance de $\sigma(x)=\sigma(y) \vee C'\sigma$.

preuve: remarquons simplement qu'il existe une substitution ρ telle que $\theta = \sigma\rho$, d'après la propriété de l'unificateur principal.

Le relèvement est analogue pour $SC2$.

4.2. Régularité sauf pour l'élément nul.

Théorème 4.7: Un ensemble de clauses S est inconsistant par rapport aux axiomes d'égalité et de régularité sauf l'élément nul si et seulement si une contradiction peut être obtenue de $S \cup \{x=x\}$ avec les règles d'inférence $CN1$, $CN2$, *O-résolution, O-paramodulation et O-factorisation.*

Definition 4.8: N-interprétation

I est une N-interprétation si c'est une E-interprétation vérifiant:

$$\forall x \quad x0 = 0$$
$$\forall x,y,z \quad (x \neq 0 \ \cap \ yx = zx) \Rightarrow y = z.$$

Définition 4.9: NT-interprétation

I est une NT-interprétation si I est une *E-interprétation* vérifiant:

condition 1: $\forall$ a, $I(a0=0)=\mathbf{V}$

condition 2: pour tous les termes a,b,c,t vérifiant les 6 conditions

1. $ac > t$
2. $ac = t$ est I-irréductible
3. c est différent de 0
4. $ac > bc$
5. $I(bc=t)= \mathbf{V}$
6. $I(ac=t)=\mathbf{F}$

nous avons $I(ac=t)= \mathbf{F}$

Lemma 4.10: I est une N-interprétation ssi c'est une NT-interprétation.

$\Rightarrow$: si $ac=t$ est I-irréductible et $I(bc=t)=\mathbf{V}$. Alors $I(c=0)=\mathbf{F}$. Si $I(ac=t)=\mathbf{V}$ alors $I(ac=bc)=\mathbf{V}$, et donc $I(a=b)=\mathbf{V}$ en contradiction avec le fait que $ac=t$ est *I-irréductible.*

$\Leftarrow$: Montrons que lorsque $I(c=0)=\mathbf{F}$ nous avons $I(ac=bc)=I(a=b)$. Sinon soit $ac=bc$ l'atome le plus petit tel que $I(c=0)=\mathbf{F}$ et $I(ac=bc) \neq I(a=b)$.

cas 1. Si $I(a=b)=\mathbf{V}$ et $I(ac=bc)=\mathbf{F}$ nous avons une contradiction car I est un modèle de la théorie de l'égalité.

cas 2: supposons que $I(a=b)=\mathbf{F}$ et $I(ac=bc)=\mathbf{V}$.

Si $I(a=a')=\mathbf{V}$ avec $a > a'$ ou $I(b=b')=\mathbf{V}$ avec $b > b'$ ou $I(c=c')=\mathbf{V}$ avec $c > c'$ nous pouvons construire un contre-exemple plus petit. Supposons donc a, b et c I-irréductibles. La preuve se termine comme dans la section précédente.

La preuve de complétude s'achève comme pour la régularité à droite. Soit I, B et J définis comme dans la section sur la régularité à droite. Considérons juste les cas où la règle qu'il faut appliquer est $CN1$ ou $CN2$. Ainsi B est supposé I-irréductible.

cas **1:** $B \equiv (a0=0)$ et $J(a0=0)=\mathbf{V}$

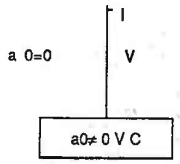


figure 5

alors une paramodulation avec x 0=0 engendre une clause falsifiée par un ancêtre de I.

cas 2: $B \equiv (ac=t)$ avec$ac > t$ et $J\,(ac=t)$= F , $ac=t \vee C$ étiquette J.
Les hypothèses entrainent $I(c=0)$=**F.**

cas 2.1: $t \equiv bc$.

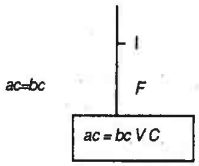


figure 6

Appliquons *CN1*, ainsi nous obtenons $a=b \vee C \vee c=0$ falsifiée par un ancêtre de I.

cas 2.2: $t \not\equiv bc$. Dans ce cas on procède comme dans la section précédente.

4.3. Régularité avec Identité.

Théorème 4.11: Un ensemble de clauses S est inconsistant par rapport aux axiomes d'égalité et de régularité avec identité si et seulement si une contradiction peut être obtenue de $S \cup \{x=x\}$ avec les règles d'inférence *CI1*, *O-résolution, O-paramodulation et O-factorisation.*

Définition 4.12: I-interprétation
I est une *I-interprétation* si c'est une *E-interprétation* vérifiant

$$\forall x, \quad ex=x$$

$$\forall x,y\,, (yx=x) \Rightarrow y=e.$$

Dans la suite nous supposons que e est le plus petit terme pour l'ordre de simplification complet défini sur l'hypothèse de Herbrand.

Définition 4.13: IT-interprétation
I est une *IT-interprétation* si c'est une *E-interprétation* vérifiant: $\forall a, c \in T(F)$
condition 1: $I\ (ea=a)$=**V.**
condition 2: $ac=c$ est I-irréductible $\Rightarrow I(ac=c)$= **F.**

Nous ne démontrons pas le lemme suivant, dont la preuve est facile.

Lemma 4.14: I est une I-interprétation ssi c'est une IT-interprétation.

Esquissons le reste de la preuve, en prenant I, B et J comme à la section précédente. Nous considérons juste les cas où B est irréductible. Avec cette hypothèse nous avons $I(a=e)$=**F.**

cas 1: $B \equiv (ea=a)$ et $J(ea=a)$=**V** et J est étiqueté par $ea \neq a \vee C$.

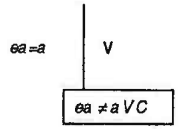


figure 7

Une paramodulation avec $ex=x$ permet de terminer.

cas 2: $B \equiv (ac=c)$ et $J\,(ac=c)$=**F** est étiqueté par $ac=c \vee C$.

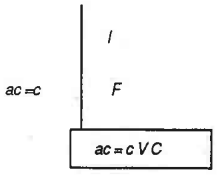


figure 8

Nous appliquons *CI1* pour obtenir $a=e \vee C$. Mais $I(a=e \vee C)$=**F** car $I(a=e)$=**F.** Nous avons ainsi une clause falsifiée par un ancêtre de I.

CONCLUSION

1. Résultats.

L'argument essentiel du travail que nous avons présenté est la notion d'*équation orientée* . Elle s'exprime dans des règles d'inférence qui sont des raffinements de la paramodulation privilégiant les membres les plus complexes des équations. Ces règles permettent de minimiser le nombre d'inférences, de simplifier les expressions et, par conséquent, de maintenir l'information sous une forme compactes et d'obtenir des preuves directes.

L'usage des équations comme règles de réécriture nécessite:

- des ordres puissants pour comparer les termes
- une technique pour prouver la complétude des stratégies fondées sur la réécriture.

En considérant la totalité des clauses apparaissant dans une dérivation, plutôt que les ensembles successifs produits par inférence, nous avons évité le raisonnement par récurrence sur les arbres sémantiques. Nous avons pu dès lors appréhender les arbres sémantiques transfinis qui poussent naturellement dès que l'on choisit d'orienter les équations avec les ordres de simplification courants de la réécriture.

Cette technique réussit à démontrer la complétude de plusieurs stratégies intéressantes de paramodulation: paramodulation ordonnée, paramodulation positive même en présence de règles de démodulation ou de subsomption, "derived reduction algorithm" de Lankford.

Nous avons interprété la procédure de complétion de Knuth et Bendix comme une stratégie de réfutation articulée sur la règle de superposition. Partant d'un ensemble de clauses de Horn consistant, l'ensemble final engendré permet de décider le problème du mot par normalisation. Nous avons ainsi prouvé indirectement la correction de la procédure de Knuth et Bendix et obtenu de nouveaux systèmes canoniques, comportant même des équations non orientables.

D'autres axiomes que l'égalité sont aussi plus efficacement traités lorsqu'il sont incorporés aux règles d'inférence. L'approche des arbres sémantiques se généralise facilement pour prouver la complétude de ces systèmes : il suffit de remplacer les interprétations de Herbrand égalitaires par des interprétations de Herbrand qui sont des modèles de la théorie envisagée.

2. Perspectives.

Il devrait être possible d'adapter notre technique de preuve pour incorporer, à la manière de Plotkin, des axiomes dans les algorithmes d'unification. Cette démarche permettra peut-être d'étendre certains résultats sur la complétion modulo une théorie équationnelle .Voir par exemple (Jouannaud Kirchner 1986).

Des travaux récents (Reddy 85) (Dershowitz Plaisted 87) (Rety et al. 1985) proposent d'utiliser la règle de surréduction, qui n'est autre que ce que nous appelons la paramodulation orientée, comme la base d'une méthode de programmation logique. Les résultats de cette thèse présentent un intérêt certain pour cette approche, puisqu'ils permettent de sortir du cadre strictement équationnel.

Une autre technique pour appliquer la réécriture à la démonstration automatique consiste à coder les formules sous une forme équationnelle grâce au système canonique booléen découvert par

J.Hsiang (1983). E.Paul (1984) a également interprété la résolution en terme de complétion équationnelle. D.Kapur et P.Narendran (1984) représentent de manière analogue les formules du premier ordre par des polynômes et utilisent la technique des bases de Grôbner (Buchberger 1976). Il semble possible d'adapter la méthode des arbres sémantiques transfinis à la preuve de complétude de ces stratégies non clausales. Cette idée est confortée par un résultat de complétude que nous avons obtenu avec J.Hsiang concernant la EN-stratégie décrite dans (Hsiang 1986). A notre connaissance, la seule stratégie fondée sur le système canonique booléen pour laquelle il existe une preuve de complétude en présence de règles de simplification est celle présentée dans (Bachmair Dershowitz 1987).

Enfin, le problème reste ouvert de savoir si la totalité des ordres de simplification utilisés est indispensable à la complétude des stratégies de paramodulation orientée.

R. Anderson, W. Bledsoe, *A Linear Format for Resolution with Merging and a New Technique for Establishing*. J.ACM 17, (1970) pp.525-534.

L. Bachmair, N. Dershowitz, J. Hsiang, *Orderings for Equational Proofs* , Symposium on Logic in Computer Science, Boston, (June 1986).

L. Bachmair, N. Dershowitz, *Inference Rules for Rewrite-based First-Order Theorem Proving* , 2nd LICS, (May 1987).

J.A. Bergstra, J.W. Klop, *Conditional Rewrite Rules: Confluency and Termination* , IW (198/82), SMC, Amsterdam, (1982).

W.W. Bledsoe, L.M. Hines, *Variable Elimination and Chaining in a Resolution-based Prover for Inequalities* , 5th CADE, LNCS #87, (1980) pp.70-87.

A.Bledsoe, K.Kunen, R.Shostak, *Completeness Results for Inequality Provers*, Artificial Intelligence, 27, 3, (1985) pp.255-288.

R. Boyer, *Locking: A Restriction of Resolution*, Ph.D. Thesis, University of Texas, Austin (1971)

R. Boyer, J.S. Moore, *A Computational Logic*, Academic Press, New York, (1979).

D. Brand, *Proving Theorems with the Modification Method* , SIAM J. of Computing, 4, (1975) pp.412-430.

D. Brand, J.A. Darringer, W.H. Joyner, *Completeness of Conditional Reductions*, Proc. 4th Conference on Automated Deduction, (1979) pp.36-42.

T. Brown, *A Structured Design Method for Specialized Proof Procedures*, Ph.D. Thesis, Cal Tech., (1974).

C.L. Chang, C.T. Lee, *Symbolic Logic and Mechanical Theorem Proving*, Academic Press (1973).

M.A.Choquer, *Preuves de formules conditionnelles dans des spécifications algébriques conditionnelles*. Thèse de 3ième cycle. Université d' Orsay. 1986.

K.L. Clark, *Negation as failure* in "Logic and Databases". eds Gallaire H. , Minker J. Plenum Press, New York 3-52 (1978).

N. Dershowitz, *Orderings for Term Rewriting Systems* , J.TCS, 17, 3, (1982) pp.279-301.

N. Dershowitz, *Applications of the Knuth-Bendix Completion Procedure*, Proc. Seminaire d'Informatique Theorique, Paris, (1983).

N. Dershowitz, *Termination*, 1st RTA, LNCS 202, (May 1985) pp.180-224.

N. Dershowitz, D. Plaisted, *Logic Programming cum Applicative Programming*, Symposium on Logic Programming, Boston, (July 1985) pp.54-67.

N. Dershowitz, D. Plaisted, *Equational Programming*, to appear in Machine Intelligence 11 (1987).

V. Digricoli, M. Harrison, *Equality-Based Binary Resolution* , J.ACM, 33, 2, (1986) pp.253-289.

T. Evans, *On Multiplicative Systems Defined by Generators and Relations* , Proc. of the Cambridge Philosophy Society, 47, (1951) pp.637-649.

F. Fages, *Formes Canoniques Dans les Algebres Booleennes, et Application a la Automatie*, Thèse, INRIA, France, (1983).

L. Fribourg, *SLOG: A Logic Programming Language Interpreter Based on Clausal Superposition*, Symposium on Logic Programming, Boston, (July 1985) pp.172-184.

L. Fribourg, *A Superposition Oriented Theorem Prover* , Theoretical Computer Science, 35, (1985) pp.129-164.

H. Gallaire, J. Minker, *Logic and Data Bases*, H. Gallaire, J. Minker, Plenum Press, New York, (1978).

H. Ganzinger, *Ground Term Confluence in Parametric Conditional Equational Specifications*, Proceeding of 4th Symposium on Theoretical Aspects of Computer Science, Passau, RFA, February 1987 .

J. Goguen, J.Meseguer, *Eqlog: Equality, Types, and Generic Modules for Logic Programming*.
J. of Logic Programming, Vol.1, Number 2, (1984) pp. 179-210.

J.A. Goguen, J.J. Tardo, *An Introduction to OBJ; A Language for Writing and Testing Formal Algebraic s*, Proceedings of the Conference on Specification of Reliable Software, Cambridge, MA 02139, (1979).

J.A. Goguen, J. W. Thatcher, E.G. Wagner, *Initial Algebra Approach to the Specification, Correctness, and Implementation of Abstract Data Types* , Current Trends in Programming Methodology, IV Data Structuring, R.T. Yeh, Prentice Hall (Automatic Computation Series), Englewood Cliffs, NJ, (1978).

J.A. Goguen, *How to Prove Algebraic Inductive Hypothesis Without Induction* , Proc. 5th Conf. on Automated Deduction, (1980) pp.356-372.

J.H. Griesmer, R.D. Jenks, *SCRATHPAD/1 - An Interactive Facility for Symbolic Mathematics* Proc. 2nd Symp. on Symbolic and Algebraic Manipulation, S. Petrick, (1971).

J. V. Guttag, J.J. Horning, *The Algebraic Specification of Abstract Data Types*, Acta Informatica, 10, 1, (1978) pp.27-52.

A.C. Hearn, *REDUCES 2 - A System and Language for Algebraic Manipulation* , Proc. 2nd Symp. on Symbolic and Algebraic Manipulation, S. Petrick , (1971)

J. Herbrand, *Recherches sur la Théorie de la Démonstration*, Travaux Soc. Sciences et Lettres Varsovie, CL.3. (Math. Phys.), 1930, 128 pp.

L. Henschen, L. Wos, *Unit Refutation and Horn Sets*, J. ACM 21(October 1974), pp. 295-301

J. Hsiang, *Refutational Theorem Proving using Term Rewriting Systems*, Artificial Intelligence, 25, (1985) pp.255-300.

J. Hsiang, *Two Results in Term Rewriting Theorem Proving* , Proc. of 1st International Conference in Rewrite Techniques and Applications, Dijon, (May 1985).

J. Hsiang, *Rewrite Method for Theorem Proving in First Order Theory with Equality* , Journal of Symbolic Computation, (1987).

J. Hsiang, N. Dershowitz, *Rewrite Methods for Clausal and Nonclausal Theorem Proving*, Proc. 10th ICALP, (July 1983) pp.331-346.

J. Hsiang, M. Rusinowitch, *On Word Problems in Equational Theories*, to appear in 14th ICALP, (1987).

J. Hsiang, M. Rusinowitch, *A New Method for Establishing Refutational Completeness in Theorem Proving* , 8th CADE, Oxford, England, (1986).

J. Hsiang, M. Rusinowitch, K. Sakai, *Complete Set of Inference Rules for the Cancellation Laws* IJCAI 87', Milan, Italy , (August 1987).

G. Huet, *Confluence Reductions: Abstract Properties and Applications to Term Rewritings* , JACM, 27, (1980) pp.797-821.

G. Huet, *A Complete Proof of Correctness of Knuth-Bendix Completion Algorithm* , JCSS, 23, (1981) pp.11-21.

G. Huet, J.M. Hullot, *Proofs by Induction in Equational Theories with Constructors* , 21st IEEE Symposium on Foundations of Computer Science, (1980) pp.797-821.

G. Huet, D.S. Lankford, *On the Uniform Halting Problem for Term Rewriting Systems*, Report 283, INRIA, (1978).

G. Huet, D.C. Oppen, *Equations and Rewrite Rules: A Survey* , Formal Languages: Perspectives and Open Problems, R. Book, Academic Press, (1980).

J.M. Hullot, *A Catalogue of Canonical Term Rewriting Systems*, Report CSL-113, SRI International, Menlo Park, CA, (1980).

H.Hussmann, *Unification in conditional equational theories*, Proc. of the EUROCAL Conference, Lect. Notes in Comp. Sci., 204

J.P.Jouannaud, B.Waldmann, *Reductive Conditional Term Rewriting Systems*, Proc. 3rd IFIP Conf. on Formal Description of Programming Concepts, Lyngby, Danemark, 1986

J.P. Jouannaud, E. Kounalis, *Automatic Proofs by Induction in Equational Theories without Constructors*, Logic in Computer Science, Boston, (1986) pp.358-366.

J.P. Jouannaud, H. Kirchner, *Completion of a Set of Rules Modulo a Set of Equations*, SIAM Journal on Computing, 15, (November 1986) pp.1155-1194.

J.P. Jouannaud, P. Lescanne, F. Reinig, *Recursive Decomposition Ordering*, Conf. on Formal Description of Programming Concepts II, D. Bjorner, North Holland, (1982) pp.331-346.

W. Joyner, *Resolution Strategies as Decision Procedures*, J. ACM, 23, (1976) pp.398-417.

S.Kaplan, *Un langage de spécification de types abstraits algébriques*, Thèse de 3ème cycle, Orsay, 1983

S.Kaplan, *Simplifying Conditional Term Rewriting Systems: Unification, Termination Confluence*, à paraitre dans J. of Symb Comp.

D. Kapur, P. Narendran, *An Equational Approach to Theorem Proving in First-Order Predicate Calculus* , 9th International Joint Conference on Artificial Intelligence, Los Angeles, CA, (August 1985).

C. Kirchner, *A New Equational Unification Method: A Generalization of Martelli-Montanari*, 7th Conference on Automated Deduction, LNCS 170, Springer-Verlag, (1984) pp.224-247.

D.E. Knuth, P.B. Bendix, *Simple Word Problems in Universal Algebra*, Computational Algebra, J. Leach, Pergamon Press, (1970) pp.263-297.

E. Kounalis, *Completeness in Data Type Specifications*, EUROCAL '85, LNCS 204, Springer-Verlag, (April 1985) pp.348-362 .

W.A. Kornfeld, *Equality in Prolog*, Proc. 8th IJCAI, Karlsruhe, Germany, (August 1983) pp.514-519.

R.A. Kowalski, *Search Strategies for Theorem Proving*, Machine Intelligence, 5, B. Meltzer, D. Michie, American Elsevier, (1970) pp.181-201.

R.A. Kowalski, *Logic for Problem Solving*, North Holland Inc., New York, NY. (1979).

R.A. Kowalski, P. Hayes, *Semantic Trees in Automatic Theorem Proving*, Machine Intelligence, 5, B Meltzer, D

Michie, American Elsevier, (1969) pp.181-201.

W. Küchlin, *A Criterion for Constraining Critical Pair Formation in Knuth-Bendix Algorithm* , Unpublished manuscript, ETH-Zentrum, (Sept. 1983).

D. Lankford, *Equality Atom Term Locking*, Ph.D. Thesis, University of Texas, Austin, TX, (1972).

D.S. Lankford, *Canonical Inference*, Report ATP-32, Univ. of Texas at Austin, (1975).

D.S. Lankford, *Some New Approaches to the Theory and Application of Conditional Term Rewriting s*, Report, Louisiana Tech Univ., (1979).

D.S. Lankford, A.M. Ballantyne, *Decision Procedure for Simple Equational Theories with Commutative-Associativity*, Report ATP-39, Univ. of TExas at Austin, (1977).

D.S. Lankford, A.M. Ballantyne, *The Refutation Completeness of Blocked Permutative Narrowing and Resolution*, 4th Conf. on Automated Deduction, Austin, TX, (1979).

D.S. Lankford, D.R. Musser, *On Semideciding First Order Validity and Invalidity*, USC-ISI Report, (1978).

P. Lescanne, *Computer Experiments with the REVE Term Rewriting System Generator* , 10th POPL, (1983).

D. Loveland, *Automated Theorem Proving: A Logical Basis*, North Holland, (1978).

D. Luckam, *Refinements in Resolution Theory*, Proceedings of the IRIA Symposium on Automatic Demonstration, New York, 1970, pp. 363-377.

E.L. Lusk, R.A. Overbeek, *A Portable Environment for Research in Automated Reasoning*, 7th Conference on Automated Deduction, Nappa Valley, CA, LNCS #170, Springer-Verlag, (1984) pp.43-52.

Z. Manna, R. Waldinger, *Special Relations in Automated Deduction*, 12th ICALP, Nafplion, Greece, (July 1985).

P. Marchand, *Cours de Logique de DEA*, Université de Nancy I (1986).

C.K. Mohan M.Srivas, *Function Definitions in Term Rewriting and Applicative Programming* , Information and Control, to appear, (1986).

J.B. Morris, *E-resolution: An Extension of Resolution to Include Equality*, IJCAI, (1969) pp.287-294.

D.R. Musser, *On Proving Inductive Properties of Abstract Data Types*, Conference Record of the Seventh Annual ACM Symposium on Principles of Progrs, Las Vegas, Nevada, (January 1980) pp.154-162.

J. Mzali, *Methodes de Filtrage Equationnel et de Preuve Automatique de Theoremes*, Thèse de Doctorat de l'Universite

de Nancy I, (1986).

P. Padawitz, *Correctness, Completeness and Consistency of Equational Data Type Specifications*, Ph.D. Thesis, Technische Universitat, Berlin, (1982).

P.Padawitz, *Horn clauses specifications: a uniform framework for abstract data types and logic programming*. Universitat Passau, MIP-8516 December 1985.

E.Paul, *Equational Methods in First Order Predicate Calculus*, Journal of Symbolic Computation, 1, (1985) pp.7-29.

E. Paul, *On Solving the Equality Problem in Theories Defined by Horn Clauses*, EUROCAL85, Linz, Austria, (April 1985) pp.363-377.

J. Pedersen, *Obtaining Complete Sets of Reductions and Equations without using Special Unification Algorithms*, Unpublished manuscript, (1985).

G. E. Peterson, *A Technique for Establishing Completeness Results in Theorem Proving with Equality* , SIAM J. of Computing, 12, 1, (1983) pp.82-100.

G.E. Peterson, M.E. Stickel, *Complete Sets of Reductions for Some Equational Theories* , JACM, 28, (1981) pp.233-264.

D.A. Plaisted, *A Recursively Defined Ordering for Proving Termination of Term Rewriting Systems* , UIUCDCS-R-78-943, Univ. of Illinois, Urbana, IL, (1978).

U. Pletat, G. Engels, H.D. Ehrich, *Operational Semantics of Algebraic Specifications with Conditional Equations*, Report 118/81, Univ. of Stuttgart, (1981).

G. Plotkin, *Building in Equational Theories*, Machine Intelligence, 7, B. Meltzer, D. Michie, American Elsevier, (1973) pp.73-90.

U.S. Reddy, *Narrowing as the Operational Semantics of Functional Languages* , Symposium on Logic Programming, Boston, (July 1985) pp.138-151.

R. Reiter, *On Closed World Data Bases*, Logic and Data Bases, H. Gallarie & J. Minker, Plenum Press, New York, pp.55-76 (1978).

R. Reiter, *Two results on Ordering for Resolution with Merging and Linear Format*, J.ACM, 18, (1971), pp. 630-646.

J.L. Rémy, *Etude des Systemes de Reecriture Conditionelle et Applications aux Types Abstraits* , Thèse d'Etat, I.N.P.L., Nancy, France, (1982).

J.L. Rémy, H. Zhang, *REVEUR4: A System for Validating Conditional Algebraic Specifications of Abss* , 5th

European Conference on Artificial Intelligence, Pisa, (1984).

J.L. Rémy, H. Zhang, *Contextual Rewriting*, Proc. of 1st International Conference in Rewrite Techniques and Applications, Dijon, (May 1985).

J.A. Robinson, *A Machine Oriented Logic based on the Resolution Principle*, JACM, 12, 1, (January 1965) pp.23-41.

J.A. Robinson, *The Generalized Resolution Principle*, Machine Intelligence, 3, D Michie, American Elsevier, (1968) pp.77-94.

K. Sakai, *Knuth-Bendix Algorithm for Thue System Based on Kachinuki Ordering*, Techincal Report, 0087, ICOT, (1985).

J. Siekmann, *Universal Unification*, 7th Conference on Automated Deduction, LNCS No 170, Springer-Verlag, (1984).

J. Slagle, *Automatic Theorem Proving with Renamable and Semantic Resolution*, J.ACM, 14, (1967) pp.687-697.

J. Slagle, *Automatic Theorem Proving For Théories with Simplifiers, Commutativity , and Associativity* J. ACM, 19 (1), (Jan. 1972) pp.120-135.

J. Slagle, *Automated Theorem Proving with Simplifiers, Commutativity, Associativity*, JACM, 21, (1974) pp.622-642.

J. Slagle, K. Norton, *Experiments with an Automatic Theorem Prover having Partial Ordering Inferens*, CACM, (1973) pp.682-688.

M. Stickel, *A Case Study of Theorem Proving by Knuth-Bendix Method Discovering that x*x*y*, 7th CADE, LNCS 170, (1984) pp.248-258.

M.E. Stickel, *A Prolog Technology Theorem Prover*, 1984 International Symposium on Logic Programming, Atlantic City, New Jersey, (Feb. 6-9 1984) pp.212-219.

R. Veroff, *Canonicalization and Demodulation*, Report ANL-81-6, Argonne National Lab, (1981).

S. Winker, L. Wos, *Procedure Implementation through Demodulation and Related Tricks* , 6th Conference on Automated Deduction, LNCS No. 125, New York, (1982) pp.109-131.

L. Wos, G.A. Robinson, *Paramodulation and Set of Support* , Symp. on Automatic Demonstration, Lecture Notes in Math. No 125, Springer-Verlag, (1970) pp.276-310.

L. Wos, G.A. Robinson, D.F. Carso, L. Shalla, *The Concept of Demodulation in Theorem Proving P*, JACM, 14, 4, (1967).

L. Wos, W. McCune, *Negative Paramodulation*, 8th CADE, LNCS #230, (1986) pp.229-239.

NOM DE L'ETUDIANT : RUSINOWITCH MICHAEL

NATURE DE LA THESE : Doctorat d'Etat ès sciences

VU, APPROUVE ET PERMIS D'IMPRIMER n° 1796.

NANCY, le 04 NOV. 1987

LE PRESIDENT DE L'UNIVERSITE DE NANCY I

RESUME

Cette thèse présente des méthodes de preuves automatique de théorèmes, favorisant l'usage des égalités comme règles de simplification, ainsi qu'une nouvelle technique pour prouver la complétude de ces méthodes. Cette technique permet de raisonner sur des *arbres sémantiques transfinis*. Nous l'avons appliquée avec succès à plusieurs stratégies de *paramodulation*, comme la paramodulation ordonnée ou la paramodulation positive.
Ces stratégies,

- n'utilisent jamais les axiomes d'égalité (sauf x=x)
- n'appliquent jamais la paramodulation dans une variable

De plus, pour accélérer les preuves, en diminuant l'espace de recherche, toutes les stratégies considérées évitent de remplacer, par paramodulation, un terme par un terme plus complexe. L'introduction des arbres sémantiques transfinis s'est justifiée pour pouvoir utiliser comme critère de comparaison des termes, des mesures de complexité d'ordinalité supérieure à ω.

La règle de *subsomption,* qui permet de supprimer les formules redondantes et la règle de *démodulation*, qui consiste à utiliser les équations dans une seule direction pour réécrire les termes en des formes plus simples, sont abondamment utilisées en démonstration automatique comme des heuristiques très efficaces (Wos, et al. 1967). En fait, l'influence de la démodulation sur la complétude des systèmes d'inférence est fort mal connue. Nous avons démontré de manière très simple, par notre méthode de preuve, que la complétude de toutes les stratégies décrites dans ce travail est conservée en présence des règles de démodulation et de subsomption.

Des auteurs ont remarqué depuis longtemps que la règle de *superposition* de l'algorithme de complétion de Knuth et Bendix n'est qu'une restriction de la paramodulation (Lankford 1975) (Brown 1974). Mais à la différence des stratégies de paramodulation, un problème essentiel de la procédure de complétion est celui de l'orientation, souvent impossible, des équations. Par extension de la règle de superposition aux équations non orientables, nous avons pu obtenir un système d'inférence réfutationnellement complet pour la logique du premier ordre avec égalité. Lorsqu'on le restreint à traiter des ensembles formés uniquement d'équations, ce système fonctionne comme une procédure de complétion. Nous obtenons donc, en prime, une nouvelle preuve de correction de l'algorithme de Knuth et Bendix (sous réserve d'utiliser un ordre de simplification). Puisque l'orientation des équations importe seulement au moment de leur utilisation, lorsqu'elles sont instanciées, notre approche permet d'obtenir des systèmes canoniques contenant des équations non orientables. Elle s'applique également avec succès à la complétion des systèmes de réécriture conditionnels.

En adaptant la notion d'interprétation de Herbrand à d'autres théories axiomatiques que l'égalité, notre méthode fournit le cadre adéquat pour la construction de systèmes de règles complets, remplaçant l'usage des axiomes. Ainsi, nous avons substitué aux axiomes de régularité (exprimant par exemple la possibilité de simplifier dans un groupe, un terme apparaissant de chaque côté d'une égalité) de nouvelles règles d'inférence, et prouvé leur complétude.

MOTS-CLEFS: démonstration automatique, résolution, paramodulation, subsomption, démodulation, algorithme de Knuth et Bendix, arbres sémantiques.