UNIVERSITE DE NANCY I
CENTRE DE RECHERCHE EN INFORMATIQUE DE NANCY

PREUVES DE TERMINAISON DES SYSTEMES DE REECRITURE ASSOCIATIFS COMMUTATIFS UNE METHODE FONDEE SUR LA REECRITURE ELLE-MEME

THESE DE DOCTORAT DE TROISIEME CYCLE EN INFORMATIQUE
PRESENTEE PAR

Isabelle GNAEDIG

SOUTENUE LE 24 FEVRIER 1986

DEVANT LA COMMISSION D'EXAMEN

PRESIDENT	R.CORI
EXAMINATEURS	F.BELLEGARDE
	J.P JOUANNAUD
	P.LESCANNE
	P.MARCHAND

UNIVERSITE DE NANCY I CENTRE DE RECHERCHE EN INFORMATIQUE DE NANCY

PREUVES DE TERMINAISON DES SYSTEMES DE REECRITURE ASSOCIATIFS COMMUTATIFS UNE METHODE FONDEE SUR LA REECRITURE ELLE-MEME

THESE DE DOCTORAT DE TROISIEME CYCLE EN INFORMATIQUE
PRESENTEE PAR

Isabelle GNAEDIG

SOUTENUE LE 24 FEVRIER 1986

DEVANT LA COMMISSION D'EXAMEN

PRESIDENT	R.CORI
EXAMINATEURS	F.BELLEGARDE
	J.P JOUANNAUD
	P.LESCANNE
	P.MARCHAND

REMERCIEMENTS

Pierre Lescanne dirige mes travaux depuis plus de deux ans. Il a su me faire partager l'interêt qu'il portait aux problèmes de la réécriture et tout particulièrement ceux de la terminaison. Je le remercie sincèrement de m'avoir guidée dans la réalisation d'un travail passionnant, et de m'avoir soutenue le long des écueuils rencontrés dans ce domaine difficile qu'est la terminaison associative-commutative.

Jean-Pierre Jouannaud m'a aidée, non seulement en ce que ma recherche s'est appuyée sur les résultats de ses travaux, mais aussi par les conseils constructifs qu'il a su me prodiguer malgré un emploi du temps extrêmement chargé. Je lui en suis très reconnaissante.

Françoise Bellegarde a suivi avec intérêt l'évolution de mon travail, et a toujours su être disponible, pour que nos discussions soient fructueuses. Aussi je la remercie chaleureusement.

Pierre Marchand a porté beaucoup d'intérêt à la réalisation de ce travail; ses remarques constructives et son soutien moral m'ont été précieux. Je lui témoigne toute ma gratitude.

Robert Cori a accepté d'être le président de mon jury. Je le remercie vivement d'avoir manifesté de l'intérêt pour mon travail.

Je tiens aussi à remercier tout particulièrement Jean-Luc Remy pour ses conseils concernant la rédaction de cette thèse, Claude et Hélène Kirchner pour leur amitié et les fructueuses discussions que nous avons eues ensemble, Jalel Mzali pour la gentillesse et la dextérité dont il a fait preuve chaque fois qu'un problème du système UNIX s'est présenté.

Merci aussi à tous les membres de l'équipe EURECA, et plus généralement du CRIN, pour leur amical soutien et leur contribution à une ambiance de travail dynamique.

Je tiens pour terminer, à manifester toute ma reconnaissance à Jacques pour son soutien de tous les instants.

TABLE DES MATIERES

INTRODUCTION

Il y a quelques décennies, des résultats algébriques furent établis pour automatiser les preuves en logique équationnelle. Church [8] , Evans [14] sont à l'origine de ce qui devait devenir une des théories les plus efficaces actuellement pour réaliser des preuves automatiques de théorèmes, et fournir une aide à la spécification de programmes: la réécriture.

Le problème essentiel en logique équationnelle est le problème du mot, qui consiste à prouver ou réfuter une égalité $t_1=t_2$, dans un contexte axiomatique donné E, les axiomes étant eux aussi sous forme d'égalités. La méthode classique de preuve consiste à déduire le théorème $t_1=t_2$ des axiomes, à l'aide des égalités axiomatiques: c'est la méthode de remplacement des égaux par des égaux. Cette méthode, classiquement utilisée par le mathématicien, a déjà été automatisée. Mais elle est inefficace, car elle nécessite de nombreux retours en arrière sur les choix d'égalités: elle est indéterministe.

L'idée de la réécriture est de limiter les choix de remplacement d'égaux par des égaux, jusqu'à rendre le processus déterministe. Elle consiste, sur l'algèbre des termes du premier ordre, à orienter les axiomes, notés sous la forme a=b, en équations à sens unilatéral ou règles de réécriture, notées $a \rightarrow b$.

Cette contrainte d'orientation constitue cependant un appauvrissement de l'ensemble d'axiomes initial. A la fin des années 1960, Knuth et Bendix proposèrent une procédure qui consiste justement à enrichir l'ensemble des

règles issues des axiomes, en ajoutant de nouvelles règles, afin d'obtenir un système de règles de réécriture équivalent à l'ensemble d'axiomes, c'est à dire ayant la même puissance de déduction [27].

Le processus de remplacement d'égaux par des égaux disparaissant au profit du processus de réécriture, il semble légitime d'exiger que ce dernier soit déterministe, c'est à dire que le résultat de la réécriture d'un terme t ne dépende pas du chemin de réécriture suivi. Cette caractéristique se traduit par une propriété intrinsèque du système de réécriture: la confluence du système.

De plus, le processus de réécriture à partir d'un terme t doit être fini, ce qui est aussi traduisible par une propriété du système de réécriture: la terminaison ou noethérianité. La dernière forme réécrite d'un terme t est alors dite forme normale de t. Si le système est confluent, la forme normale de t est unique. La procédure de Knuth et Bendix fournit en effet, lorsqu'elle se termine avec succès, un système confluent et noethérien, à partir d'un ensemble d'axiomes.

Le test de validité d'une équation $t_1=t_2$ dans une théorie équationnelle donnée E se fait alors par normalisation. Il consiste à réécrire les termes t_1 et t_2 avec le système de réécriture équivalent aux axiomes de E, jusqu'à ne plus pouvoir appliquer de règle. La propriété de Church et Rosser exprime que si $t_1=t_2$ est valide dans E, alors il existe un t tel que t_1 et t_2 se réécrivent en t. Pour un système confluent et noethérien, le théorème $t_1=t_2$ est valide dans E si et seulement si les formes normales $t_1\downarrow$ et $t_2\downarrow$ de t_1 et t_2 sont égales. L'algorithme de Knuth et Bendix constitue donc une procédure de décision pour le problème du mot

dans les théories équationnelles (Entscheidungsproblem).

Il est important de noter que la confluence est une propriété globale du système, décrite par l'interaction des règles entre elles. Dans les années 1940, Newman a établi que pour un système de réécriture noethérien, la confluence équivalait à une propriété aisément vérifiable sur le système: la confluence locale [32]. Pour tester cette nouvelle propriété, on introduit la notion de paire critique, qui traduit qu'un membre gauche de règle peut se réécrire à l'aide d'une autre règle du système. On voit ici apparaître le double intérêt de la propriété de terminaison, devant à la fois interdire un processus infini de réécriture et permettre la preuve de confluence.

Pour la preuve de noethérianité, l'outil le plus usuellement employé est la notion d'ordre noethérien sur les termes, incluant la relation de réécriture. Moyennant certaines propriétés d'un tel ordre, on pourra garantir qu'un système est noethérien si, pour chaque règle, le membre gauche est supérieur au membre droit.

La terminaison étant un problème indécidable, il n'existe pas d'algorithme universel, ni même d'ordre général sur les termes permettant de prouver la noethérianité des systèmes de réécriture. On peut en contrepartie développer des outils de preuve dans des cas particuliers et des contextes précis de la réécriture.

L'objet de cette thèse est de développer un ordre permettant la preuve de terminaison dans certains cas de réécriture équationnelle associative commutative.

La notion de réécriture équationnelle est née du fait que certaines équations ne doivent pas être orientées en règles, faute de quoi le système devient obligatoirement non noethérien. C'est le cas pour l'axiome de commutativité qui, s'il est orienté en $x+y \rightarrow y+x$, engendre la réécriture infinie $x+y \rightarrow y+x \rightarrow x+y \ldots$ Le moyen de traiter de tels axiomes est donc de les garder sous forme d'équations, et d'adapter la théorie de la réécriture en considérant une égalité modulo cet ensemble d'axiomes E, partout où la réécriture simple considérait une égalité stricte. Le problème du mot en réécriture équationnelle consistera non plus à regarder si $t_1{\downarrow} = t_2{\downarrow}$, mais si $t_1{\downarrow} =_E t_2{\downarrow}$. La réécriture équationnelle sera définie non plus par une chaîne de réécritures $t_1 \rightarrow t_2 \rightarrow t_3 \ldots$, mais par une chaîne alternée de réécritures et d'égalités $t_1 =_E t_2 \rightarrow t_3 =_E \ldots$ La terminaison de la réécriture équationnelle sera la non existence d'une chaîne infinie du type $t_1 =_E t_2 \rightarrow t_3 =_E \ldots \rightarrow t_n =_E \ldots$

Nous nous placerons dans le contexte où l'ensemble des axiomes non orientables E est un ensemble d'axiomes de commutativité $f(x,y)=f(y,x)$ et d'associativité $f(f(x,y),z)=f(x,f(y,z))$. Ce contexte algébrique est souvent rencontré et largement utilisé. Il est de plus difficile à éviter. L'axiomatisation des entiers est en effet fondée sur des opérations associatives et commutatives telles que + et *. Donnons aussi l'exemple usuel des groupes abéliens définis par

$$0+x \rightarrow x$$

$$-x+x \rightarrow 0$$

$$--x \rightarrow x$$

$$-0 \rightarrow 0.$$

Pour expliquer le caractère technique du travail qui va suivre, soulignons que l'apparente simplicité et la familiarité des problèmes associatifs-commutatifs masquent un champ de recherche dense et délicat. Nous citons, à titre d'exemple, le problème de Syracuse [1] , système apparemment simple dont personne n'a encore pu prouver la terminaison sur les entiers strictement positifs:

$$f(x+x) \rightarrow f(x)$$

$$f(x+x+1) \rightarrow f(x+x+x+1+1)$$

où + est une opération associative et commutative.

Soulignons enfin le caractère universel de la théorie de la réécriture depuis les mathématiques jusqu'à l'informatique la plus appliquée. Le problème du mot en a été la motivation historique, mais il est remarquable que cette théorie ait sous-tendu des principes suffisamment puissants pour qu'on lui découvre plus tard d'autres champs d'application tout aussi importants. En effet, outre les preuves de théorèmes équationnels, elle permet les preuves de théorèmes par récurrence. Elle utilise l'"induction sans induction" dont le principe est le suivant: sous certaines conditions relatives à la notion de constructeur, le théorème non équationnel est ajouté à l'ensemble des règles. Si la procédure de complétion appliquée à cet ensemble fournit un système confluent et noethérien, alors le théorème est inductivement valide.

La réécriture fournit aussi un outil de validation des types abstraits algébriques, ce en quoi elle est utile à tout informaticien manipulant un langage de programmation typé.

Elle s'utilise aussi comme interprète des langages de programmation, par la traduction des programmes en ensembles d'équations. On voit poindre ici l'impact de la réécriture dans les secteurs appliqués de l'intelligence artificielle où, dans un futur proche, elle sera amenée à compléter des langages de type clausal comme PROLOG.

Des applications ont de plus été proposées pour la vérification des réseaux de Pétri [7] , et pour les tests de dépendances dans les bases de données [9]. Notons aussi l'utilisation des systèmes de réécriture d'expressions fonctionnelles comme outils de transformation de programmes itératifs [4].

Plusieurs implantations de la procédure de Knuth et Bendix ont été réalisées, qui fournissent de véritables laboratoires de réécriture, et permettent de l'exploiter dans tous ses champs d'application. Citons, sans être le moins du monde exhaustive, le laboratoire RRLAB du projet SEKI [38] , le système interactif ERIL de J.Dick [13] , FORMEL [15] , et le logiciel REVE. Ce projet est développé au sein d'une coopération internationale constituée autour des équipes EURECA du CRIN à Nancy, et SPD au MIT, à Cambridge, USA. Après une version expérimentale REVE1 écrite par P. Lescanne durant un séjour au MIT [29] , une version REVE2 s'est développée pour satisfaire en particulier des exigences de modularité et d'interface avec l'utilisateur [16].

Dejà motivée par l'importance de la preuve de terminaison, j'ai réalisé, dans le cadre de mon DEA, une automatisation dans REVE1, du processus d'orientation des règles, réduisant l'interaction avec l'utilisateur au cours de la complétion [17]. Puis a été écrit, dans la lignée de REVE2,

le laboratoire de réécriture conditionnelle REVEUR4 [36] , et le laboratoire REVE3 de réécriture équationnelle, en particulier associative-commutative [23] , [26]. Celui-ci n'offre pas, jusqu'ici, de garantie effective de terminaison pour les systèmes qu'il traite, les ordres utilisés n'étant pas adaptés au cas équationnel. En effet, l'intérêt suscité par ce problème étant assez nouveau, les seuls travaux publiés sont nécessairement récents et encore incomplets. La motivation du travail qui suit est donc de réaliser un outil de preuve implantable dans REVE3, en l'étayant de justifications théoriques.

Situons alors notre approche dans le contexte des travaux déjà effectués dans ce domaine. Dans [2], Bachmair et Dershowitz définissent des conditions sur la réécriture, pour prouver la terminaisons des systèmes équationnels. Ces conditions ont été établies en utilisant une approche indépendante de la nôtre. A ce titre, ils sont plutôt complémentaires.

Par ailleurs, il existe une méthode classique pour prouver la terminaison d'un système de réécriture, qui consiste à utiliser un ordre bien fondé sur les termes, incluant la relation de réécriture. Pour un aperçu des ordres usuellement employés, on pourra consulter [12].

Dans [6] , A. Ben Cherifa et P. Lescanne proposent un ordre fondé sur l'interprétation des termes par des polynômes. Ils caractérisent exactement les polynômes qui peuvent être utilisés pour la preuve de terminaison des systèmes de réécriture associatifs-commutatifs. Leur approche est également complémentaire de la nôtre.

Bachmair & Plaisted proposent un ordre de simplification fondé sur une transformation des termes qui les "aplatit" et distribue leurs opérateurs

les uns par rapport aux autres. Ils imposent en outre une restriction sur la précédence: la condition sur les paires [35] , [3].

Présentons maintenant le principe de notre approche. Le point de départ du présent travail est un ensemble de théorèmes établis par Jouannaud et Muñoz. Ceux-ci introduisent des restrictions sur les ordres utilisés en terminaison simple (c.a.d. $E = \emptyset$), afin qu'ils deviennent applicables au cas équationnel (c.a.d. $E \neq \emptyset$) [24]. Nous nous proposons de construire un ordre qui satisfasse ces restrictions, dans le cas de la théorie équationnelle associative-commutative (notée AC). Cet ordre est fondé sur une interprétation [28] des termes de l'algèbre libre T(F,X), par des termes de l'"algèbre des termes aplatis" TV(F,X), et nous semble une approche plus naturelle que celle de Huet et Oppen, qui interprète les termes par des polynômes [19]. Notre approche est différente de celle de Bachmair et Plaisted sous divers aspects. Nous essayons d'être plus précis dans la définition des transformations inhérentes à la méthode. C'est pourquoi nous nous imposons un seul axiome de distributivité; ce qui n'est pas une réelle restriction car les exemples usuels en terminaison AC ne contiennent qu'un axiome de distributivité. D'autre part, nous différencions et séparons strictement les concepts d'"aplatissement" et de distributivité. En effet, l'"aplatissement" permet l'interprétation des termes "purs" par des termes "aplatis", assurant à l'ordre la propriété de AC-commutation. Quant à la distributivité, elle projette les termes dans la même classe distributive, rendant l'ordre F-compatible. La AC-commutation est essentielle et des ordres tels que le RPO seul, qui ne satisfont pas cette propriété, ne fournissent pas de preuve valide de la terminaison AC [5]. De plus, nous proposons une preuve tout à fait générale d'une

propriété de l'ordre relative aux termes non clos: la stabilité par instanciation.

En outre, cet ordre, qui a été conçu pour traiter des systèmes orientables avec le mécanisme de distributivité, peut être étendu à des ordres construits sur le même type de règles, comme l'endomorphisme. Dans ce cas, nous proposons un ordre fondé sur le même principe, en remplaçant la règle de distributivité par une règle d'endomorphisme. De la même façon, nous établissons que cet ordre est aussi un ordre de réduction AC-commutant, et généralisons le principe à d'autres règles de transformation.

Esquissons le plan de lecture du travail qui va suivre. Dans le Chapitre 1, nous établissons le contexte algébrique de la réécriture, et présentons les principales définitions utilisées ultérieurement. Dans le Chapitre 2, nous exposons le problème de la terminaison en général, et donnons un aperçu des différentes méthodes déjà existantes pour établir cette propriété. Le Chapitre 3 décrit la construction de notre ordre AC, et nous prouvons que cet ordre est un ordre de réduction AC-commutant dans le Chapitre 4. Le cas des variables est résolu dans le Chapitre 5. Le Chapitre 6 donne des exemples d'application de l'ordre construit, en cernant les limites de son utilisation. Dans le Chapitre 7, nous présentons une extension de la théorie à d'autres règles annexes que la distributivité. Dans la conclusion, nous faisons le bilan du travail effectué, et exposons les perspectives et prolongations que cette étude a suggérées.

CHAPITRE 1: ALGEBRE ET REECRITURE

1. Le contexte algébrique

Donnons maintenant une description formelle du cadre algébrique dans lequel s'inscrit ce travail.

Définition 1 : Soit F un ensemble énumérable de symboles de fonctions et $a:F\rightarrow N$ une fonction appelée arité. Soit [A] un ensemble. Une F-algèbre A sur [A] est une paire $([A],F_A)$, où $F_A = \{f_A \mid f\in F\}$ est un ensemble d'opérations définies sur [A], tel que si $a(f)=n$, alors f_A est une opération de $[A]^n$ dans [A].

Définition 2 : Soient $A=([A],F_A)$ et $B=([B],F_B)$ deux algèbres. Un homomorphisme h de A dans B est une application de [A] dans [B], telle que pour tout symbole de fonction f dans F avec $a(f)=m$, pour $a_1,\ldots a_m$ dans [A], nous ayons:

$h(f_A(a_1,\ldots a_m)) = f_B(h(a_1),\ldots h(a_m))$.

Si de plus, h est bijective, alors c'est un isomorphisme.

Définissons alors la notion d'algèbre libre.

Définition 3 : Soient K une classe quelconque de F-algèbres et X un ensemble. Une F-algèbre K-libre engendrée par X est une F-algèbre $([A],F_A)$ telle que

1) $([A],F_A)$ est dans K

2) [A] contient X

3) pour toute algèbre $([B],F_B)$ de K et toute application $h_0:X\rightarrow[B]$, il

existe un homomorphisme unique $h:[A]\rightarrow[B]$ tel que la restriction de h à X soit h_0.

Soit K la classe de toutes les F-algèbres pour F donné, et X un ensemble de variables. La K-algèbre libre engendrée par X existe. C'est l'algèbre des termes construite sur F et X, que nous noterons T(F,X).

2. La réécriture

Nous supposons que le lecteur connaît les notions usuelles de sous-terme, variable, terme clos, occurrence et substitution. Pour un rappel de ces définitions et des notations correspondantes, nous renvoyons à [19].

Nous appellerons équation une paire t=t' de termes de T(F,X). La théorie équationnelle induite par un ensemble E d'équations est la plus petite congruence $=_E$ de T(F,X), contenant les axiomes de E et fermée par substitution.

Nous appellerons système de réécriture R de T(F,X) tout ensemble fini de paires ordonnées $g\rightarrow d$, où g et d sont dans T(F,X), et telles que $V(d) \subseteq V(g)$. V(t) est l'ensemble des variables du terme t.

La relation de réécriture induite par R est la plus petite relation $\rightarrow_R$ de T(F,X), contenant R et close pour les opérations de l'algèbre et par substitution. La relation $-+\rightarrow_R$ (resp. $-*\rightarrow_R$) est la fermeture transitive (resp. réflexive-transitive) de $\rightarrow_R$. La relation $\rightarrow_R$ est dite confluente si et seulement si, pour tous termes s, t, t', tels que $s -*\rightarrow_R t$ et $s -*\rightarrow_R t'$, il existe un terme s' tel que $t -*\rightarrow_R s'$ et $t' -*\rightarrow_R s'$.

Nous appellerons système de réécriture équationnel toute paire (R,E)

où R est un système de réécriture, et E un ensemble d'équations. La relation de réécriture R/E est définie par $=_E \cdot \rightarrow_R \cdot =_E$.

Notre but ici est de prouver, dans un cas particulier de E, la E-terminaison d'un système de réécriture R, définie comme suit:

Définition 4 : Un système de réécriture R termine modulo E (ou est noethérien modulo E) si et seulement s'il n'existe pas de séquence infinie de la forme $t_1 =_E t_1' \rightarrow_R t_2 =_E t_2' \rightarrow_R \ldots =_E t_n' \rightarrow_R \ldots$. Ce qui signifie aussi que la relation R/E termine. Si E = ø, on dira que R termine (ou est noethérien).

CHAPITRE 2: TERMINAISON

1. Problèmes généraux et méthodes de preuves

Comme nous l'avons vu, la théorie de la réécriture et la procédure de complétion de Knuth et Bendix ne sont valides et exploitables que si la propriété de terminaison est vérifiée: en l'absence de preuve de terminaison, on ne peut parler de preuve automatique.

De par l'indéterminisme de la réécriture, il existe plusieurs concepts de terminaison, traduisant des phénomènes différents. Un système de réécriture peut être tel qu'à partir d'un terme quelconque, il existe toujours au moins une chaîne de dérivations finie, donc une forme normale: c'est la terminaison faible. Mais il peut être à la fois difficile de trouver cette forme normale parmi toutes les possibilités de réécriture, et fâcheux d'avoir des dérivations infinies.

On introduit donc le concept de terminaison forte ou terminaison uniforme, qui garantit que toute chaîne de dérivations partant d'un terme donné est finie. Nous ne retiendrons que cette deuxième notion: elle seule permet les applications de la réécriture au problème du mot. Tous les outils développés pour la preuve de terminaison concernent la terminaison uniforme; nous utiliserons désormais le terme terminaison dans ce sens.

Se pose alors le problème de la décidabilité de la terminaison. De l'indécidabilité de l'arrêt des machines de Turing, on a déduit l'indécidabilité de la terminaison des systèmes de réécriture [18]. Remarquons toutefois que le problème est décidable dans l'algèbre initiale

(l'algèbre des termes clos).

Il faudra donc se contenter de conditions suffisantes pour prouver la terminaison. L'outil le plus employé est la notion d'ordre bien-fondé sur les termes (pour lequel il n'existe pas de chaîne infinie descendante), et contenant la relation de réécriture (tel que si $s \rightarrow t$ alors $s > t$). L'existence d'un tel ordre garantit la noethérianité de la relation de réécriture de façon évidente. Supposons en effet qu'on ait une chaîne de dérivations infinie $s_1 \rightarrow s_2 \rightarrow \ldots s_m \rightarrow \ldots$. On obtient donc $s_1 > s_2 > \ldots s_m > \ldots$, ce qui contredit l'hypothèse.

Manna et Ness ont proposé une méthode permettant d'établir facilement qu'un ordre contient la relation de réécriture, en introduisant la notion de compatibilité de l'ordre avec les opérations de l'algèbre $T(F,X)$.

Définition 5 : Un ordre $>$ sur $T(F,X)$ est dit F-compatible si et seulement si pour tous s et t de $T(F,X)$ tels que $s > t$, on a $f(\ldots s \ldots) > f(\ldots t \ldots)$.

Définition 6 : Un ordre sur $T(F,X)$ est un ordre de réduction si et seulement s'il est bien-fondé et F-compatible.

Théorème 1 : [Manna et Ness]: Un système de réécriture est noethérien si et seulement s'il existe un ordre de réduction $>$ tel que pour toute règle $g \rightarrow d$ du système et toute substitution σ, on ait $\sigma g > \sigma d$.

La méthode de Manna et Ness présente cependant un désavantage: il est souvent malaisé de prouver la bonne-fondation d'un ordre sur les termes. Une notion plus complète a été introduite par Dershowitz [10] , pour éviter cette preuve de bonne-fondation: c'est la notion d'ordre de simplification. Elle requiert la propriété de sous-terme, plus forte que la bonne-

fondation, et aisément vérifiable. Elle exprime qu'un terme est toujours plus grand qu'un quelconque de ses sous-termes.

Définition 7 : Un ordre sur T(F,X) est un ordre de simplification si et seulement s'il possède les trois propriétés suivantes:

$s>t \Rightarrow f(...s...) > f(...t...)$ (F-compatibilité)

$f(...s...) > s$ (sous-terme)

$f(...s...) > f(... ...)$ (effacement)

On notera que la condition d'effacement n'est nécessaire que dans une algèbre de termes où les symboles n'ont pas une arité fixe.

Théorème 2 : [Dershowitz]: Un système de réécriture de termes comportant un nombre fini de symboles de fonctions est noethérien, s'il existe un ordre de simplification > tel que pour toute règle $g \rightarrow d$ du système et toute substitution σ, on ait $\sigma g > \sigma d$.

Notons qu'une caractéristique essentielle des ordres de simplification est de contenir un ordre de simplification minimal: l'ordre de plongement noté ∇.

Définition 8 : On dit qu'un terme $s=f(...s_i...)$ est plongé dans un terme $t=g(...t_j...)$, et on note $s \nabla t$ si et seulement si:

(1) il existe j tel que s est plongé dans t_j

(2) f=g et pour tout i, $s_i \nabla t_i$

Exemple 1 : Le terme f(x,g(y,z)) est plongé dans le terme f(h(x,z),g(h(x,y),h(y,z))).

Le principe de preuve de terminaison exprimé par le théorème de

Dershowitz repose sur la notion de plongement et sur le théorème de Kruskall.

Théorème 3 : [Kruskall]: Soit une F-algèbre (F fini). Dans toute séquence infinie de termes t_1, t_2, ...t_n..., il existe une paire de termes t_i, t_j avec i<j et telle que $t_i \nabla t_j$.

La preuve du théorème de Dershowitz est alors immédiate. Supposons qu'il existe une séquence infinie de termes $t_1 \rightarrow t_2 \rightarrow \ldots t_n \rightarrow \ldots$ et donc $t_1 > t_2 > \ldots > t_n \ldots$. D'après le théorème de Kruskall, il existe i et j tels que j>i et $t_i \nabla t_j$. Or comme l'ordre > contient le plongement, $t_j > t_i$, ce qui contredit l'hypothèse.

Divers ordres de simplification ont été proposés et sont maintenant largement utilisés. Ils sont pour la plupart construits à partir d'un ordre sur les symboles de fonctions appelé précédence. On citera l'ordre sur les chemins de sous-termes [34] , l'ordre récursif sur les chemins ou RPO [10,25] , l'ordre récursif de décomposition ou RDO [21,30,37].

Les ordres RPO et RDO ont été implantés dans le logiciel REVE, fournissant un outil automatique de preuve de terminaison. On utilise en effet la puissance du théorème de terminaison de Dershowitz en réduisant l'indéterminisme dû à la condition sur les substitutions. Pour ce faire, on exige une propriété supplémentaire des ordres de simplification: la stabilité par instanciation.

Définition 9 : Un ordre sur T(F,X) est stable par instanciation si et seulement si, pour s et t tels que s>t, on a $\sigma s > \sigma t$, quelle que soit la substitution σ.

Il est alors clair que pour un ordre de simplification stable par instanciation >, la condition suffisante de terminaison du théorème de Dershowitz peut se réduire à: pour toute règle $g \rightarrow d$ du système, on a $g > d$. Le RPO et le RDO sont des ordres stables par instanciation.

Définissons alors l'un des ordres les plus employés: le RPO. Outre la notion de précédence, cet ordre utilise la notion de multiensemble, et d'ordre sur les multiensembles.

Définition 10 : Un multiensemble M sur un ensemble E est une application de E dans l'ensemble des entiers naturels N. Le domaine D(M) est l'ensemble des x de E tels que $M(x) \neq 0$.

Exemple 2 : E = N et M = {3,5,8,8,8}

M(3)=1 M(5)=1 M(8)=3

Tout ordre de E peut facilement être étendu à un ordre sur les multiensembles sur E. Il existe plusieurs façons de définir l'ordre multiensemble. Nous donnons ici la version de Huet et Oppen.

Définition 11 : Etant donné un ordre > sur un ensemble E, l'ordre multiensemble >> sur MU(E), l'ensemble des multiensembles sur E, est défini par:

M>>N si et seulement si $M \neq N$ et (N(y)>M(y) => il existe x dans E tel que x>y et M(x)>N(x))

Exemple 3 : E = N U {a,b,c,d} avec 1<2<3... et a<b<c...

alors {2,3,4,4,b,c,c,d,d} >> {1,3,3,4,a,d,d}

Donnons alors la définition de l'ordre RPO sur les termes.

Définition 12 : Soit $>_F$ une précédence sur F. Supposons que toutes les

variables de X sont incomparables entre elles et avec les symboles de F. Soient s et t deux termes de T(F,X). L'ordre récursif sur les chemins ou RPO noté $>_{RPO}$ est défini récursivement comme suit: $s=f(s_1,...s_m) >_{RPO} t=g(t_1,...t_n)$ si et seulement si:

$s >_{RPO} t_j$ pour j=1,...n et

soit f=g et $\{s_1,...s_m\} >>_{RPO} \{t_1,...t_n\}$

soit $f >_F g$

soit $s_i \geq RPO\ t$,ce qui signifie $s_i >_{RPO} t$ ou $s_i \simeq t$ pour un i dans [1,m], où $\simeq$ désigne la congruence de permutation dans T(F,X).

Exemple 4 : Prouvons la terminaison d'un système de règles définissant la suite de Fibonacci:

$$fib(0) \rightarrow 0$$

$$fib(s(0)) \rightarrow s(0)$$

$$fib(s(s(x))) \rightarrow +(fib(x),fib(s(x))).$$

Par propriété sous-terme du RPO, on a $fib(0) >_{RPO} 0$ et $fib(s(0)) >_{RPO} s(0)$. Avec la précédence $fib >_F +$, on obtient $fib(s(s(x))) >_{RPO} +(fib(x),fib(s(x)))$. En effet, $fib(s(s(x))) >_{RPO} fib(x)$ car $s(s(x)) >_{RPO} x$ et $fib(s(s(x))) >_{RPO} fib(s(x))$ car $s(s(x)) >_{RPO} s(x)$.

2. Terminaison équationnelle

Le même problème de terminaison se pose dans le cas de la réécriture équationnelle. Cette propriété est ici aussi un point clé de la théorie de la réécriture, conditionnant toutes les propriétés importantes qui permettent ses applications.

Remarquons que le problème de la terminaison, comme nous l'avons

défini dans le chapitre précédent, équivaut au problème de la terminaison de la relation R/E. Cette réécriture est malheureusement indécidable dans le cas général. Afin de la rendre automatisable, on va introduire une relation $\rightarrow_R$, décidable vérifiant $\rightarrow_R \subseteq \rightarrow_R, \subseteq \rightarrow_{R/E}$, puis des propriétés permettant de calculer avec $\rightarrow_R$, de façon équivalente à $\rightarrow_{R/E}$: la E-cohérence et la E-confluence.

Dans le cas de systèmes E-noethériens, la preuve conjointe des deux propriétés précédentes est décidable: elle se ramène à des tests sur les paires critiques étendus à R et E, à condition que la relation $\rightarrow_R$, ait une valeur particulière. Plusieurs solutions ont été proposées pour R': la relation $\rightarrow_R$ elle-même si R est linéaire gauche [20] , la relation $\rightarrow_{R,E}$ [33] , un mélange des deux [22].

A partir de l'étude de l'ensemble des relations $\rightarrow_R$, vérifiant $\rightarrow_R \subseteq \rightarrow_R, \subseteq \rightarrow_{R/E}$, Jouannaud et Muñoz ont établi que la E-terminaison de $\rightarrow_R$ se réduisait à la terminaison simple de $\rightarrow_R$, pourvu que $\rightarrow_R$, vérifie la propriété de E-commutation. Ce résultat a permis l'adaptation de la méthode de Manna et Ness au cas équationnel, moyennant la E-commutation de la relation d'ordre [31].

Définition 13 : Une relation > est E-commutante si et seulement si pour tous s, t, s' de T(F,X) tels que $s' =_E s > t$, il existe un t' dans T(F,X) tel que $s' > t' =_E t$.

Théorème 4 : [Muñoz]: Soit (R,E) un système de réécriture équationnelle défini sur T(F,X). (R,E) est noethérien s'il existe un ordre de réduction E-commutant qui oriente toutes les instances de R.

Remarquons que de façon analogue, on peut obtenir une généralisation au cas équationnel de la notion d'ordre de simplification (voir aussi Muñoz).

3. La terminaison AC

Un contexte théorique de la terminaison équationnelle ayant été établi, des analogies avec les méthodes de Manna et Ness et Dershowitz ayant été faites, nécessité s'est fait sentir d'automatiser les preuves de E-terminaison. On a donc essayé de construire des ordres de E-réduction et de E-simplification, tout particulièrement dans le cas où E est la théorie équationnelle associative-commutative.

Donnons tout d'abord la définition de la théorie AC. Soit (F_{AC}, F_{NAC}) une partition de l'ensemble F de symboles. L'ensemble E=AC est l'ensemble des équations de la forme f(x,y) = f(y,x), f(x,f(y,z)) = f(f(x,y),z) pour tout f de F_{AC} (ou en notation infixée x+y = y+x et x+(y+z) = (x+y)+z)). La théorie associative commutative AC est la théorie équationnelle induite par AC.

Le problème de la terminaison AC est réputé difficile et n'a été abordé que récemment. Un premier article [11] a tenté de faire le point sur les ordres pouvant être adaptés au cas AC. Puis Plaisted [35] a proposé un ordre de simplification fondé sur la notion de transformation de termes par distributivité et aplatissement, et qui a été le point de départ du présent travail.

Notre but ici est de construire un ordre de réduction AC-commutant, utilisant ainsi le théorème de Muñoz pour fournir à REVE3 un processus

automatique de preuve de terminaison AC. Contrairement au travail de Bachmair et Plaisted cité dans l'introduction, notre ordre n'est pas un ordre de simplification mais un odre de réduction: la propriété de noethérianité est ici beaucoup plus simple à démontrer que la propriété de sous-terme. D'autre part, la transformation des termes par distributivité et aplatissement sont pour nous complètement distinctes. Nous mettrons en effet en évidence le fait qu'on "normalise" les termes par distributivité, puis que dans un deuxième temps seulement, on aplatit ces formes normalisées. Cette méthode permet l'utilisation de concepts simples; seule la distributivité sur les symboles binaires est utilisée: $x*(y+z) \rightarrow (x*y)+(x*z)$. Ceci n'est pas le cas chez Bachmair et Plaisted, qui intercallent les étapes de distributivité et d'aplatissement, ce qui nécessite une infinité de règles distributives de la forme:

$$*(x_1,\ldots x_m,+(y_1,\ldots y_n)) \rightarrow +(*(x_1,\ldots x_m,y_1),\ldots *(x_1,\ldots x_m,y_n)).$$

De plus, nous proposons une approche générale du cas des variables, en prouvant la stabilité par instanciation de l'ordre pour des substitutions quelconques.

CHAPITRE 3: CONSTRUCTION D'UN ORDRE AC

1. Un ordre AC-commutant

Nous proposons ici de construire un ordre qui satisfasse les conditions du théorème de Muñoz, lorsque E = AC. Il semble naturel de partir d'un ordre de réduction déjà connu, tel que le RPO, ou le RDO qui est un ordre un peu plus puissant que le RPO. Nous fixerons notre choix sur le RPO pour la simple raison que sa définition est d'un abord plus simple que celle du RDO.

Cet ordre ne convient pourtant pas au cas associatif-commutatif, car il ne possède pas la propriété de AC-commutation. En effet, si $s >_{RPO} t$ et $s =_{AC} s'$, il n'existe pas forcément un t' tel que $s' >_{RPO} t' =_{AC} t$. A titre d'exemple, il suffit de prendre s = (x+y)+z, t = f(x+y), et s' = x+(y+z), avec + ≃ f pour la précédence.

Nous allons donc forcer la AC-commutation de l'ordre, et ce en projetant les termes à comparer sur un représentant particulier de leur classe d'équivalence AC, avant d'effectuer la comparaison avec le RPO. Un représentant d'un terme peut être obtenu par "aplatissement" de ce terme suivant ses symboles AC (ce qui traduit les axiomes d'associativité), et par l'attribution d'un statut multiensemble aux symboles AC (ce qui traduit les axiomes de commutativité). Donc, si $s =_{AC} t$, les représentants des deux termes sont équivalents par permutation ≃ suivant leurs symboles AC. Le RPO utilisé sera le RPO multiensemble originel, défini sur l'algèbre des termes variadiques TV(F,X).

Afin de donner une définition précise de la projection d'un terme sur un représentant de sa classe AC, définissons la fonction arité variable av sur l'ensemble F des symboles, qui induira le concept d'algèbre des termes variadiques.

Définition 14 : Soit P(N) l'ensemble de tous les sous-ensembles de N. La fonction $av:F\rightarrow P(N)$ est telle que si f est dans F_{NAC}, alors $av(f)=\{a(f)\}$, et si f est dans F_{AC} alors $av(f) = N-\{0,1\}$.

Définition 15 : L'algèbre TV(F,X) est l'algèbre libre engendrée par X, où la fonction arité sur F est av. Les éléments de TV(F,X) sont appelés termes variadiques.

Remarque : Notons que $T(F,X) \subseteq TV(F,X)$.

Définition 16 : L'opération d'aplatissement est une application de TV(F,X) dans TV(F,X), telle que pour tout terme s de TV(F,X), la forme aplatie de s notée [s] est la forme normale de s pour un système infini décrit par le schéma de règle suivant:

$f(y_1,..y_{i-1},f(x_1,..x_m),y_{i+1},..y_n) \rightarrow f(y_1,..y_{i-1},x_1,..x_m,y_{i+1},..y_n)$

pour chaque f de F_{AC}, quel que soit i>0, quels que soient m et n >1.

Remarque : La forme normale de s existe et est unique. Le système précédent est en effet localement confluent et noethérien. Notons qu'à un terme donné, seul un sous-ensemble fini de l'ensemble des règles s'applique, d'où décidabilité de la réductibilité.

Remarquons de plus que pour tout sous-terme $f(s_1,...s_m)$ d'un terme aplati [s], où f est dans F_{AC}, le symbole de tête des $s_1,...s_m$ ne peut être f. Nous noterons $s_1+s_2+...s_{m-1}+s_m$ pour $+(s_1,...s_m)$. L'ensemble des termes

aplatis est un sous-ensemble de l'ensemble des termes varyadiques. Il peut être doté d'une structure de F-algèbre. Notons FL(F,X) l'algèbre des termes aplatis.

Explicitons alors le lien entre les deux relations d'équivalence $=_{AC}$ et $\simeq$. L'algèbre T(F,X) est partitionnée en classes d'équivalences AC définies par la relation $=_{AC}$. Par définition de la relation $\simeq$, il apparait que deux termes commutativement égaux sont équivalents par permutation des sous-termes suivant leurs symboles AC. D'autre part, deux termes associativement égaux ont la même forme aplatie dans l'algèbre des termes aplatis FL(F,X). Par conséquent, l'opération d'aplatissement permet d'interpréter les classes AC de T(F,X) par les classes $\simeq$ de FL(F,X). Si $s =_{AC} t$, alors $[s] \simeq [t]$ dans l'algèbre FL(F,X), ou $\dot{[s]}=\dot{[t]}$ (où $\dot{[s]}$ désigne la classe de s pour la relation $\simeq$) dans l'algèbre $FL(F,X)/\simeq$. Nous travaillerons donc sur les algèbres $FL(F,X)/\simeq$ et $TV(F,X)/\simeq$. Plus exactement, nos raisonnements seront fondés sur des termes représentants des classes d'équivalence $\simeq$ plutôt que sur les classes elles-mêmes.

2. Un ordre compatible avec les opérations de l'algèbre

Considérons maintenant la terminaison AC d'un système de réécriture contenant la règle de distributivité

$$x * (y+z) \rightarrow (x*y) + (x*z).$$

Avec le RPO, nous obtenons $x*(y+z) >_{RPO} (x*y)+(x*z)$, en imposant pour la précédence $* >_F +$. Cette condition sur la précédence définit un ordre $>$ qui n'est pas F-compatible. En effet, $x*(y+z) > (x*y)+(x*z)$, mais $u*(x*(y+z)) < u*((x*y)+(x*z))$, car $[u*(x*(y+z))] = u*x*(y+z)$, $[u*((x*y)+(x*z))] = u*((x*y)+(x*z))$, et $u*((x*y)+(x*z)) >_{RPO} u*x*(y+z)$.

Par conséquent, nous devons forcer la F-compatibilité de l'ordre >. Pour ce faire, nous sommes guidés par l'idée intuitive de réduire aussi le terme majoré lorsque le terme majorant est réduit par aplatissement. Nous introduisons dans cette optique une transformation supplémentaire sur les termes, qui sera la réécriture par la règle de distributivité elle-même. Cette transformation est une réduction, car le RPO muni de la précédence $*>_F+$ oriente la règle de distributivité. Notons que la transformation distributive trie les occurences du symbole + par rapport au symbole * dans le terme: toutes les occurences de + "montent" vers la racine, alors que les occurences de * descendent vers les feuilles.

Soient à comparer les deux termes z*(x*y) et z*(x+y). Le premier terme s'aplatit en z*x*y, le second devient (z*x)+(z*y) par réécriture avec la règle de distributivité. Nous obtenons $z*x*y >_{RPO} (z*x)+(z*y)$, ce qui préserve la F-compatibilité de l'ordre >.

Remarquons que pour obtenir de manière symétrique (x*y)*z > (x+y)*z, nous devons introduire la règle de distributivité symétrique $(x+y)*z \rightarrow (x*z)+(y*z)$ dans la transformation.

Définissons alors de façon formelle la transformation distributive.

Définition 17 : L'opération de distributivité est une transformation de T(F,X) dans T(F,X), telle que pour tout terme s de T(F,X), une forme distribuée de s, est obtenue en réécrivant s en une forme irréductible avec le système (D) défini par les deux règles suivantes:

$$x*(y+z) \rightarrow (x*y)+(x*z) \qquad (1)$$

$$(x+y)*z \rightarrow (x*z)+(y*z). \qquad (2)$$

En effet, pour tout s de T(F,X), il existe une forme irréductible car le système précédent est noethérien. Cette forme irréductible n'est cependant pas unique car (D) n'est pas confluent. Nous ne pouvons donc, dans ce cas, parler de forme normale (nous emploierons toutefois la notation $\downarrow$ attribuée habituellement aux formes normales).

Par exemple, le terme (x+y)*(u+v) se réécrit en ((x*u)+(x*v))+((y*u)+(y*v)) en utilisant (2) puis (1) deux fois, et en ((x*u)+(y*u))+((x*v)+(y*v)) en utilisant (1) puis (2) deux fois. Nous pouvons toutefois énoncer la propriété suivante.

Propriété : Si $s -*\rightarrow_D s'$ et $s -*\rightarrow_D s''$ et s' et s" sont irréductibles, alors $[s'] \simeq [s"]$.

Nous sommes alors en mesure de définir la transformation complète d'un terme s de T(F,X) par distributivité et aplatissement.

Définition 18 : L'opération de distributivité et aplatissement est une transformation de la sous-algèbre T(F,X) de TV(F,X) dans FL(F,X), telle qu' une forme distribuée aplatie d'un terme s de T(F,X) est une forme irréductible $[s\downarrow]$, où $s\downarrow$ est une forme irréductible de s pour (D).

Remarque : D'après la propriété précédente, toutes les formes $[s\downarrow]$ sont équivalentes par permutation $\simeq$ sur FL(F,X) (autrement dit, leurs classes dans $FL(F,X)/\simeq$ sont égales).

Remarque : Pour deux formes irréductibles s' et s" (pour D) du même terme s, $s' >_{RPO} t \iff s" >_{RPO} t$, car le RPO est compatible avec la congruence de permutation.

Des deux remarques précédentes, nous déduisons que la comparaison de deux termes sur leur formes D-irréductibles aplaties par le RPO ne dépend pas de la forme D-irréductible de chaque terme, qui est unique à la congruence de permutation près. Par abus de notation, $[s\downarrow]$ désigne non pas une forme unique, mais une des formes permutativement équivalentes.

Notons que la transformation précédente doit être considérée comme l'interprétation dans TV(F,X), par aplatissement, d'une forme D-irréductible $s\downarrow$ de s. Donnons alors la définition de l'ordre.

Définition 19 : Soient s et t deux termes de T(F,X). Nous dirons que s>t si et seulement si

(1) $[s\downarrow] >_{RPO} [t\downarrow]$ ou

(2) $[s\downarrow] \simeq [t\downarrow]$ et $s \text{ -+}\!\!\rightarrow_{D/AC} t$ où D désigne ici la règle $x*(y+z)\rightarrow(x*y)+(x*z)$.

Remarque : Dans le cas (2) de la définition précédente, une seule des deux règles symétriques de distributivité suffit, de par la définition de la relation $\rightarrow_{D/AC}$.

Les symboles + et * désigneront désormais les symboles AC qui vérifient $*>_F+$. Soulignons en effet que nous n'autorisons qu'une seule paire de symboles AC comparables, ce qui nécessite pour l'ordre une seule paire de règles distributives: les règles de distributivité du symbole * par rapport au symbole +.

Remarquons que nous permettons à deux termes ayant le même représentant associatif-commutatif-distributif (ACD) d'être comparables si l'un des termes se réécrit en l'autre. Ce nous semble un critère simple et

naturel, et permet par exemple d'orienter un axiome tel que: $x*(y+z) = (x*z)+(x*y)$. En effet, $x*(y+z) \ -+\rightarrow_{D/AC} \ (x*z)+(x*y)$. Donc $x*(y+z) > (x*z)+(x*y)$. Evidemment, la relation $-+\rightarrow_{D/AC}$ induite par la règle de distributivité D est un ordre sur $T(F,X)$, car la fermeture transitive d'une relation de réécriture noethérienne est un ordre.

Exemple 5 : Soient à comparer les deux termes $s=(f(x)+y)*(z+t)$ et $t=(x*z)+t$ par le NFLO. On a $(f(x)+y)*(z+t) \rightarrow_D (f(x)*(z+t))+(y*(z+t)) \rightarrow_D ((f(x)*z)+(f(x)*t))+((y*z)+(y*t))$. Et $[((f(x)*z)+(f(x)*t))+((y*z)+(y*t))] = (f(x)*z)+(f(x)*t)+(y*z)+(y*t)$. D'autre part, $[t\downarrow] = t = (x*z)+t$. Alors $s>t$ car $(f(x)*z)+(f(x)*t)+(y*z)+(y*t) >_{RPO} (x*z)+t$.

CHAPITRE 4: UN ORDRE DE REDUCTION AC-COMMUTANT

1. La relation NFLO est AC-commutante

Prouvons tout d'abord que notre relation >, que nous nommerons NFLO, possède la propriété de AC-commutation; et même qu'elle satisfait la propriété plus forte de AC-complétude.

Définition 20 : Une relation > est AC-complète si et seulement si, pour tous s, t, s', t' de T(F,X) tels que $s' =_{AC} s > t =_{AC} t'$, on a $s' > t'$.

Théorème 5 : La relation NFLO est AC-complète.

Preuve

Par hypothèse, s' =AC s > t =AC t'. D'autre part, $s' =_{AC} s$ implique $[s'\downarrow] \simeq [s\downarrow]$ et $t' =_{AC} t$ implique $[t'\downarrow] \simeq [t\downarrow]$.

Si $[s\downarrow] >_{RPO} [t\downarrow]$, alors par compatibilité de la congruence de permutation $\simeq$ avec le RPO, $[s'\downarrow] >_{RPO} [t'\downarrow]$ c.a.d. s'>t'.

Si $[s\downarrow] \simeq [t\downarrow]$, alors $[s'\downarrow] \simeq [t'\downarrow]$. De plus, $s -+\!\!\rightarrow_{D/AC} t$ signifie que $s =_{AC} s'' -+\!\!\rightarrow_{D/AC} t'' =_{AC} t$. Donc $s' =_{AC} s'' -+\!\!\rightarrow_{D/AC} t'' =_{AC} t'$ c.a.d. $s' -+\!\!\rightarrow_{D/AC} t'$. □

2. La relation NFLO est un ordre de réduction

Abordons maintenant les propriétés relatives à un ordre de réduction. L'irréflexivité, la transitivité et la bonne fondation de la relation NFLO découlent directement de l'irréflexivité, la transitivité et la bonne fondation des deux relations RPO et $-+\!\!\rightarrow_{D/AC}$, et de la fonctionnelle lexi-

cographique qui préserve les ordres bien fondés. **La relation > est donc un ordre bien fondé.**

Soit alors à prouver la F-compatibilité de l'ordre. Rappelons que pour s>t, on doit avoir h(...s...) > h(...t...). Nous allons établir cette propriété en la décomposant en plusieurs lemmes suivant la valeur du symbole h.

2.1. Les cas simples de F-compatibilité

Considérons tout d'abord le cas (2) de la définition de l'ordre, pour lequel la F-compatibilité est quasi-immédiate.

Lemme 1 : Dans le cas où $[s\downarrow] \simeq [t\downarrow]$ et $s \text{ -+}\rightarrow_{D/AC} t$, pour tout h de F, on a: s>t => h(...s...) > h(...t...).

Preuve

Si $s \text{ -+}\rightarrow_{D/AC} t$ alors, par définition de la relation de réécriture, $h(...s...) \text{ -+}\rightarrow_{D/AC} h(...t...)$. Donc $[h(...s...)\downarrow] \simeq [h(...t...)\downarrow]$. □

Nous considérerons désormais le seul cas où s>t signifie que $[s\downarrow] >_{RPO} [t\downarrow]$.

Lemme 2 : Si h est dans F_{NAC}, alors s>t => h(...s...) > h(...t...).

Preuve

Remarquons tout d'abord que $[h(...s...)\downarrow] = h(...[s\downarrow]...)$ et $[h(...t...)\downarrow] = h(...[t\downarrow]...)$, où les termes du contexte de $h(...[s\downarrow]...)$ et $h(...[t\downarrow]...)$ sont respectivement les formes distribuées aplaties des termes du contexte de h(...s...) et h(...t...) (termes notés ...). Par

hypothèse, s>t soit $[s\downarrow] >_{RPO} [t\downarrow]$. Le résultat découle immédiatement de la F-compatibilité du RPO. □

2.2. Le RPO et l'opération d'aplatissement

Nous allons développer ici une propriété très importante du RPO vis à vis du processus d'aplatissement des termes, afin de pouvoir traiter les cas de F-compatibilité où le symbole h est dans F_{AC}. Nous allons établir que le RPO est compatible avec l'opération d'aplatissement, à condition toutefois que les symboles AC suivant lesquels on aplatit soient minimaux pour la précédence. Introduisons la notion d'aplatissement suivant un seul symbole AC.

Définition 21 : La forme "k-aplatie" $[s]_k$ de s est la forme normale de s pour le système infini décrit par le schéma de règle suivant:

$k(y_1,..y_{i-1},k(x_1,..x_m),y_{i+1},..y_n) \rightarrow k(y_1,..y_{i-1},x_1,..x_m,y_{i+1},..y_n)$

quel que soit i>0, quels que soient m et n >1.

Donnons alors le lemme exprimant la compatibilité du RPO avec l'opération d'aplatissement.

Lemme 3 : Si k est dans F_{AC} et est minimal dans F, alors $s>_{RPO}t \Rightarrow [s]_k >_{RPO} [t]_k$.

Preuve

Soient $s=f(s_1,...s_m)$ et $t=g(t_1,...t_n)$. La preuve se fait par induction structurelle sur l'algèbre T(F,X). On distinguera plusieurs cas suivant les formes $[s]_k$ et $[t]_k$.

1. $[s]_k=f([s_1]_k,...[s_m]_k)$ et $[t]_k=g([t_1]_k,...[t_n]_k)$

Distinguons alors les cas de la définition du RPO, utilisés pour comparer les termes s et t.

Si f=g et $\{s_1,\ldots s_m\} >>_{RPO} \{t_1,\ldots t_n\}$, alors par hypothèse d'induction, $\{[s_1]_k,\ldots[s_m]_k\} >>_{RPO} \{[t_1]_k,\ldots[t_n]_k\}$. Donc $[s]_k >_{RPO} [t]_k$.

Si f>g et si $s >_{RPO} t_j$ pour tout j dans [1,n], alors par hypothèse d'induction, $[s]_k >_{RPO} [t_j]_k$ pour tout j dans [1,n]. Donc $[s]_k >_{RPO} [t]_k$.

S'il existe i dans [1,n] tel que $s_i \geqq_{RPO} t$, alors, par hypothèse d'induction, $[s_i]_k \geqq_{RPO} [t]_k$. Donc $[s]_k >_{RPO} [t]_k$.

2. $[\underline{s}]_k \neq \underline{f}([s_1]_k,\ldots[s_m]_k)$ et $[\underline{t}]_k = \underline{g}([t_1]_k,\ldots[t_n]_k)$

Dans ce cas, f=k et il existe i tel que $[s_i]_k = k([s_{i1}]_k,\ldots[s_{ip}]_k)$. Donc $[s]_k = k([s_1]_k,\ldots[s_{i1}]_k,..[s_{ip}]_k,..[s_m]_k)$.

2.1 g=k et $\{s_1,\ldots s_m\} >>_{RPO} \{t_1,\ldots t_n\}$

S'il existe j tel que $s_i >_{RPO} t_j$, alors par hypothèse d'induction, $[s_i]_k >_{RPO} [t_j]_k$ c.a.d. $k([s_{i1}]_k,\ldots[s_{ip}]_k) >_{RPO} [t_j]_k$. D'autre part, le symbole de tête de $[t_j]_k$ ne peut être k. Donc, il existe s_{ih} tel que $[s_{ih}]_k \geqq_{RPO} [t_j]_k$, car k est minimal dans F. Donc $\{[s_1]_k,..[s_{i1}]_k,..[s_{ip}]_k,..[s_m]_k\} >>_{RPO} \{[t_1]_k,\ldots[t_n]_k\}$ c.a.d. $[s]_k >_{RPO} [t]_k$.

Si $t_i,\ldots t_n$ ne sont pas majorés par s_i, alors par hypothèse d'induction, on a toujours: $\{[s_1]_k,..[s_{i1}]_k,..[s_{ip}]_k,..[s_m]_k\} >>_{RPO} \{[t_1]_k,\ldots[t_n]_k\}$ c.a.d. $[s]_k >_{RPO} [t]_k$.

2.2 g≠k

Dans ce cas, il existe s_j tel que $s_j \geq_{RPO} t$, car k est minimal dans F.

Si j=i alors $s_j = k(s_{i1},\ldots s_{ip})$. Par hypothèse d'induction, $[s_j]_k \geq_{RPO} [t]_k$ c.a.d. $k([s_{i1}]_k,\ldots[s_{ip}]_k) \geq RPO\ [t]_k$. De plus, il existe s_{ih} tel que $[s_{ih}]_k \geq_{RPO} [t]_k$, car k est minimal dans F. Donc, $k([s_1]_k,..[s_{i1}]_k,..[s_{ip}]_k,..[s_m]_k) >_{RPO} [t]_k$.

Si j≠i alors, par hypothèse d'induction, $[s_j]_k \geq_{RPO} [t]_k$. Donc, $[s]_k >_{RPO} [t]_k$ car $[s_j]_k$ est un sous-terme de $[s]_k$.

3. $[\underline{s}]_k = \underline{f}([s_1]_k,\ldots[s_m]_k]_k$ et $[\underline{t}]_k \neq \underline{g}([t_1]_k,\ldots[t_n]_k)$

Dans ce cas, g=k et il existe j tel que $[t_j]_k = k([t_{j1}]_k,\ldots[t_{jq}]_k)$. Donc $[t]_k = k([t_1]_k,..[t_{j1}]_k,..[t_{jq}]_k,..[t_n]_k)$.

Si f=k et $\{s_1,\ldots s_m\} >>_{RPO} \{t_1,\ldots t_n\}$, alors $\{[s_1]_k,\ldots[s_m]_k\} >>_{RPO} \{[t_1]_k,\ldots[t_n]_k\}$ par hypothèse d'induction. De plus, $[t_j]_k >_{RPO} [t_{j1}]_k,\ldots[t_{jq}]_k$ par propriété sous-terme du RPO. Donc $\{[s_1]_k,\ldots[s_m]_k\} >>_{RPO} \{[t_1]_k,..[t_{j1}]_k,..[t_{jq}]_k,..[t_n]_k\}$ c.a.d. $[s]_k >_{RPO} [t]_k$.

Si f>k et si pour tout j dans [1,n], $s >_{RPO} t_j$, alors par hypothèse d'induction, $[s]_k >_{RPO} [t_j]_k$ pour tout j. Comme dans le cas précédent, $[t_j]_k >_{RPO} [t_{j1}]_k,\ldots[t_{jq}]_k$. Donc $[s]_k >_{RPO} [t_{j1}]_k,\ldots[t_{jq}]_k$. Donc $[s]_k >_{RPO} [t]_k$.

S'il existe i dans [1,m] tel que $s_i \geq_{RPO} t$, et f≠k, alors par hypothèse d'induction, $[s_i]_k \geq_{RPO} [t]_k$. Donc $[s]_k >_{RPO} [t]_k$ car $[s_i]_k$ est un sous-terme de $[s]_k$.

4. $[\underline{s}]_k \neq \underline{f}([s_1]_k,\ldots[s_m]_k)$ et $[\underline{t}]_k \neq \underline{g}([t_1]_k,\ldots[t_n]_k)$

Dans ce cas, f=g=k et $[s]_k = k([s_1]_k,..[s_{i1}]_k,..[s_{ip}]_k,..[s_m]_k)$ et $[t]_k = k([t_1]_k,..[t_{j1}]_k,..[t_{jq}]_k,..[t_n]_k)$. Par hypothèse, $\{s_1,...s_m\} >>_{RPO} \{t_1,...t_n\}$.

Pour les $s_h >_{RPO} t_l$ avec $h \neq i$ et $l \neq j$, on a $[s_h]_k >_{RPO} [t_l]_k$ par hypothèse d'induction.

Pour les t_l tels que $s_i >_{RPO} t_l$ avec $l \neq j$, remarquons que le symbole de tête de s_i est k, et que le symbole de tête de t_l ne peut être k. Par hypothèse d'induction, $[s_i]_k >_{RPO} [t_l]_k$. Donc il existe s_{ih} tel que $[s_{ih}]_k \geq_{RPO} [t_l]_k$. En effet, le symbole de tête de s_i est k, k est minimal dans F, et $[s_i]_k = k([s_{i1}]_k,...[s_{ip}]_k)$.

Pour les $s_h >_{RPO} t_j$ avec $h \neq i$, par hypothèse d'induction, $[s_h]_k >_{RPO} [t_j]_k$. Par propriété sous-terme du RPO, on obtient $[sh]_k >_{RPO} [t_{j1}]_k,...[t_{jq}]_k$.

Pour $s_i >_{RPO} t_j$, remarquons que le symbole de tête de s_i et de t_j est k. Par hypothèse d'induction, $[s_i]_k >_{RPO} [t_j]_k$. Donc $\{[s_{i1}]_k,...[s_{ip}]_k\} >>_{RPO} \{[t_{j1}]_k,...[t_{jq}]_k\}$.

Considérant les quatre cas précédents, on obtient en définitive $\{[s_1]_k,..[s_{i1}]_k,..[s_{ip}]_k,..[s_m]_k\} >>_{RPO} \{[t_1]_k,..[t_{j1}]_k,..[t_{jq}]_k,..[t_n]_k\}$ c.a.d. $[s]_k >_{RPO} [t]_k$. □

La condition de minimalité de k est primordiale dans le lemme précédent. En effet, supposons qu'il existe un symbole f dans F tel que $k >_F f$. Dans ce cas, $k(x,k(y,z)) >_{RPO} k(x,f(y,z))$. Or $[k(x,k(y,z))]_k = k(x,y,z)$ et $k(x,y,z)$ non$>_{RPO} k(x,f(y,z))$.

2.3. Les cas litigieux de F-compatibilité

Utilisant le lemme précedent, nous allons maintenant exprimer la F-compatibilité du NFLO lorsque les termes à comparer sont enracinés par un symbole h, AC et minimal.

Lemme 4 : Si $k \neq *$ est dans F_{AC} et est minimal dans F, alors $s>t \Rightarrow k(s,u) > k(t,u)$.

Preuve

On a $[k(s,u)\downarrow] = [k([s\downarrow],[u\downarrow])]_k$ et $[k(t,u)\downarrow] = [k([t\downarrow],[u\downarrow])]_k$. Par hypothèse, $[s\downarrow] >_{RPO} [t\downarrow]$. Par F-compatibilité du RPO, $k([s\downarrow],[u\downarrow]) >_{RPO} k([t\downarrow],[u\downarrow])$. Le résultat est donné par le lemme précédent. □

Exemple 6 : On a $x*y > x+y$ car $*>+$. On a encore $(x*y)+z > (x+y)+z$ car $[(x*y)+z\downarrow] = (x*y)+z$, $[(x+y)+z\downarrow] = x+y+z$ et $(x*y)+z >_{RPO} x+y+z$.

Exemple 7 : On a $x+f(y) > f(y)$ sans hypothèse de précédence. On a encore $(x+f(y))+z > f(y)+z$ car $[(x+f(y))+z\downarrow] = x+f(y)+z$ et $x+f(y)+z >_{RPO} f(y)+z$.

Remarque : La notation $[s*u\downarrow]$ signifie qu'on réécrit par D toute l'expression contenue entre le crochet ouvrant [et le signe $\downarrow$ (ici $s*u$). Si la réécriture par D ne porte que sur le dernier terme de l'expression soit u, on notera $[s*(u\downarrow)]$.

Nous allons maintenant établir une propriété qui nous permettra de prouver le dernier lemme de F-compatibilité, où l'enracinement des termes à comparer se fait avec le symbole *.

Lemme 5 : Si u n'est ni de la forme u_1+u_2, ni de la forme u_1*u_2, et si $[s\downarrow]$

est de la forme $[s_1\downarrow]+[s_2\downarrow]+\ldots+[s_m\downarrow]$, où $s_1,s_2,\ldots s_m$ sont dans T(F,X), alors $[s*u\downarrow]$ est de la forme $[s_1*u\downarrow]+[s_2*u\downarrow]+\ldots+[s_m*u\downarrow]$.

Preuve

Par hypothèse, il existe $a_1,\ldots a_m$ dans FL(F,X) tel que $[s\downarrow] = a_1+a_2+\ldots+a_m$. D'autre part, $s\downarrow =_{AC} s_1+(s_2+\ldots+(s_{m-1}+s_m)\ldots)$ où $s_1,s_2,\ldots s_m$ sont dans T(F,X) et $a_1=[s_1]$, $a_2=[s_2],\ldots a_m=[s_m]$. Donc,

$$\begin{aligned} s*u\downarrow &=_{AC} ((s_1+(s_2+\ldots+(s_{m-1}+s_m)\ldots))*u)\downarrow \\ &=_{AC} ((s_1*u)+((s_2*u)+\ldots+((s_{m-1}*u) + (s_m*u))\ldots))\downarrow \\ &=_{AC} s_1*u\downarrow+(s_2*u\downarrow+\ldots+(s_{m-1}*u\downarrow+s_m*u\downarrow)\ldots). \end{aligned}$$

Donc $[s*u\downarrow] = [s_1*u\downarrow]+\ldots+[s_m*u\downarrow]$, car le symbole de tête des s_i ne peut être +. □

Lemme 6 : Si + est minimal dans F, et s'il n'existe pas $f\neq+$ tel que $*>_F f$, alors $s>t \Rightarrow s*u > t*u$.

Preuve

La preuve se fait par induction structurelle sur l'algèbre T(F,X), utilisant les lemmes 3 et 5. On distinguera plusieurs cas suivant la forme de u, s, et t.

1 $u = u_1+u_2$

Dans ce cas, $[s*u\downarrow] = [s*(u_1+u_2)\downarrow] = [[s*u_1\downarrow]+[s*u_2\downarrow]]_+$. De la même façon, $[t*u\downarrow] = [[t*u_1\downarrow]+[t*u_2\downarrow]]_+$. Par hypothèse d'induction, $[s*u_1\downarrow] >_{RPO} [t*u_1\downarrow]$ et $[s*u_2\downarrow] >_{RPO} [t*u_2\downarrow]$. Donc $[s*u_1\downarrow]+[s*u_2\downarrow] >_{RPO} [t*u_1\downarrow]+[t*u_2\downarrow]$ d'après la définition de l'ordre RPO. D'après le lemme 3, $[[s*u_1\downarrow]+[s*u_2\downarrow]]_+ >_{RPO} [[t*u_1\downarrow]+[t*u_2\downarrow]]_+$.

2 $u = u_1 * u_2$

Par hypothèse, s>t, ce qui implique, par induction, que $s*u_1 > t*u_1$ et $(s*u_1)*u_2 > (t*u_1)*u_2$. Donc $s*(u_1*u_2) > t*(u_1*u_2)$ car > est AC-commutant.

3 $u = k(u_1,\ldots u_p)$

3.1 $[s\downarrow]$ est de la forme $[s_1\downarrow]+\ldots+[s_m\downarrow]$, où $s_1,\ldots s_m$ sont dans $T(F,X)$

D'après le lemme 5, $[s*u\downarrow]$ est de la forme $[s_1*u\downarrow]+\ldots+[s_m*u\downarrow]$.

3.1.1 $[t\downarrow]$ est de la forme $[t_1\downarrow]+\ldots+[t_n\downarrow]$, où $t_1,\ldots t_n$ sont dans $T(F,X)$

D'après le lemme 5, $[t*u\downarrow]$ est de la forme $[t_1*u\downarrow]+\ldots+[t_n*u\downarrow]$. Par hypothèse, s>t, soit $[s\downarrow] >_{RPO} [t\downarrow]$, c.a.d. $\{[s_1\downarrow]\ldots[s_m\downarrow]\} >>_{RPO} \{[t_1\downarrow]\ldots[t_n\downarrow]\}$. Par hypothèse d'induction, $\{[s_1*u\downarrow]\ldots[s_m*u\downarrow]\} >>_{RPO} \{[t_1*u\downarrow]\ldots[t_n*u\downarrow]\}$ c.a.d. $[s*u\downarrow] >_{RPO} [t*u\downarrow]$.

3.1.2 $[t\downarrow]$ n'est pas de la forme $[t_1\downarrow]+\ldots+[t_n\downarrow]$

Par hypothèse, $[s\downarrow] >_{RPO} [t\downarrow]$. Puisque + est minimal dans F, il existe i tel que $[s_i\downarrow] \geq_{RPO} [t\downarrow]$. Par hypothèse d'induction, $[s_i*u\downarrow] \geq_{RPO} [t*u\downarrow]$, donc $[s*u\downarrow] >_{RPO} [t*u\downarrow]$.

3.2 $[s\downarrow]$ est de la forme $[s_1\downarrow]*\ldots*[s_m\downarrow]$

3.2.1 $[t\downarrow]$ est de la forme $[t_1\downarrow]+\ldots+[t_n\downarrow]$,

Puisque $*>_F+$, $[s\downarrow] >_{RPO} [t_j\downarrow]$ pour tout j dans [1,n]. Par hypothèse d'induction, $[s*u\downarrow] >_{RPO} [t_j*u\downarrow]$. Donc $[s*u\downarrow] >_{RPO} [t*u\downarrow]$.

3.2.2 $[t\downarrow]$ est de la forme $[t_1\downarrow]*\ldots*[t_n\downarrow]$,

La preuve se fait comme dans le cas 3.1.1, en remplaçant + par *.

3.2.3 $t=f(t_1,\ldots t_n)$ avec $f\neq *$ et $f\neq +$

Dans ce cas, $[s*u\downarrow]$ est de la forme $[s_1\downarrow]*\ldots*[s_m\downarrow]*[u\downarrow]$, et $[t*u\downarrow]$ est de la forme $[t\downarrow]*[u\downarrow]$. On a $* \text{ non}>_F f$. Donc il existe i tel que $[s_i\downarrow] \geq_{RPO} [t\downarrow]$. Donc, d'après la définition des multiensembles, $\{[s_1\downarrow],\ldots,[s_m\downarrow],[u\downarrow]\} >>_{RPO} \{[t\downarrow],[u\downarrow]\}$, c.a.d. $[s*u\downarrow] >_{RPO} [t*u\downarrow]$.

3.3 $s=g(s_1,\ldots s_m)$ avec $g\neq *$ et $g\neq +$

Dans ce cas, $[s\downarrow]$ est de la forme $g([s_1\downarrow],\ldots[s_m\downarrow])$ et $[s*u\downarrow] = [s\downarrow]*[u\downarrow]$.

3.3.1 $[t\downarrow]$ est de la forme $[t_1\downarrow]+\ldots+[t_n\downarrow]$

Rappelons que $[t*u\downarrow]$ est de la forme $[t_1*u\downarrow]+\ldots+[t_n*u\downarrow]$. Par hypothèse, $[s\downarrow] >_{RPO} [t\downarrow]$. Par propriété sous-terme du RPO, $[s\downarrow] >_{RPO} [t_j\downarrow]$ pour tout j dans [1,n]. Par hypothèse d'induction, $[s*u\downarrow] >_{RPO} [t_j*u\downarrow]$, c.a.d. $[s\downarrow]*[u\downarrow] >_{RPO} [t_j*u\downarrow]$. Donc $[s*u\downarrow] >_{RPO} [t*u\downarrow]$ car $*>_F+$.

3.3.2 $[t\downarrow]$ est de la forme $[t_1\downarrow]*\ldots*[t_n\downarrow]$

Par hypothèse, $[s\downarrow] >_{RPO} [t\downarrow]$, et par propriété sous-terme du RPO, $[s\downarrow] >_{RPO} [t_j\downarrow]$ pour tout j dans [1,n]. Donc, $\{[s\downarrow], [u\downarrow]\} >>_{RPO} \{[t_1\downarrow],\ldots[t_n\downarrow],[u\downarrow]\}$. Donc $[s*u\downarrow] >_{RPO} [t*u\downarrow]$.

3.3.3 $t= f(t_1,\ldots t_n)$ avec $f\neq *$ et $f\neq +$

Dans ce cas, $[t*u\downarrow] = [t\downarrow]*[u\downarrow]$. Par hypothèse, $[s\downarrow] >_{RPO} [t\downarrow]$. Par F-compatibilité du RPO, $[s\downarrow]*[u\downarrow] >_{RPO} [t\downarrow]*[u\downarrow]$. □

Notons l'importance des conditions sur la précédence du lemme 6. Sup-

posons en effet qu'il existe un symbole f≠+ tel que $*>_F f$. On a x*y > f(x). Mais la propriété de F-compatibilité est trahie car (x*y)*z non> f(x)*z. En effet, [((x*y)*z)↓] = x*y*z et $x*y*z \text{ non}>_{RPO} f(x)*z$.

Nous donnons maintenant le théorème général exprimant la propriété de F-compatibilité du NFLO.

Théorème 6 : Sous les conditions de précédence suivantes:

*>+

si k et k' sont dans F_{AC}, $k>_F k'$ => k=* et k'=+

si k est dans F_{AC} et k ≠ *, alors k est minimal dans F

si f est dans F, $*>_F f$ => f=+

l'ordre NFLO est F-compatible.

Preuve

La preuve découle immédiatement des lemmes 1, 2, 4, et 6.

CHAPITRE 5: LE CAS DES VARIABLES

Rappelons les conditions exigées par le théorème de Muñoz, pour la preuve de terminaison d'un système de réécriture. Un système de réécriture est noethérien modulo AC, si pour toute règle $g \rightarrow d$ du système, et toute substitution σ, $\sigma g > \sigma d$. La condition précédente est malheureusement indécidable en général. Le but du présent travail étant de fournir un outil de preuve implantable, essayons de modifier cette condition. Nous introduisons alors une propriété plus forte, la stabilité par instanciation, qui exprime le comportement de l'ordre > avec l'opération de substitution.

Rappel: Un ordre > sur les termes est stable par instanciation si et seulement si $s>t \Rightarrow \sigma s>\sigma t$ pour toute substitution σ.

1. Substitutions et formes irréductibles aplaties.

Citons alors quelques propriétés évidentes, afin d'établir la stabilité par instanciation de l'ordre NFLO.

Lemme 7 : Pour tout terme s de T(F,X) et toute substitution σ, on a $[\sigma s\downarrow] = [\sigma\ [s\downarrow]\downarrow]$

Définition 22 : La forme D-irréductible aplatie d'une substitution σ est la substitution notée $[\sigma\downarrow]$ définie par $[\sigma\downarrow]\ x = [\sigma x\downarrow]$.

Notons la non unicité de $[\sigma\downarrow]$, vu que la forme D-irréductible aplatie de chaque instanciation de variable est définie à la congruence de permutation près.

Lemme 8 : Pour tout terme s de T(F,X) et toute substitution σ, on a $[\sigma s\downarrow] = [[\sigma\downarrow]\ s\downarrow]$.

2. Les cas simples de stabilité par instanciation

Comme pour la propriété de F-compatibilité, le cas (2) de la définition du NFLO se traite de façon immédiate.

Lemme 9 : Si $s>t$ avec $[s\downarrow] \simeq [t\downarrow]$ et $s \text{-+}\!\!\rightarrow_{D/AC} t$, alors $\sigma s>\sigma t$.

Preuve

D'après les lemmes 7 et 8, $[\sigma s\downarrow] = [[\sigma\downarrow]\ [s\downarrow]\downarrow]$. De la même façon, $[\sigma t\downarrow] = [[\sigma\downarrow]\ [t\downarrow]\downarrow]$. Donc $[s\downarrow] \simeq [t\downarrow]$ implique $[\sigma s\downarrow] \simeq [\sigma t\downarrow]$. D'autre part, par définition de la relation de réécriture, si $s \text{-+}\!\!\rightarrow_{D/AC} t$ alors $\sigma s \text{-+}\!\!\rightarrow_{D/AC} \sigma t$. Donc $\sigma s>\sigma t$. □

Nous considérerons désormais le seul cas où $s>t$ signifie que $[s\downarrow] >_{RPO} [t\downarrow]$.

Définition 23 : Une substitution est dite élémentaire si et seulement si son domaine est réduit à une seule variable.

Lemme 10 : Soit σ une substitution close de domaine $DOM = \{x_1,\ldots x_n\}$. Alors $\sigma = \sigma_1\ldots\sigma_n$ où σ_i est la substitution close élémentaire de domaine $\{x_i\}$.

Nous allons prouver la stabilité par instanciation du NFLO sur les substitutions closes, puis nous montrerons que cette propriété est équivalente à la stabilité par instanciation générale, sur les substitutions quelconques. Exprimons alors sous forme de lemmes les différents cas de stabilité par instanciation suivant la forme de σ. Soit σ une substitu-

tion close élémentaire de domaine {x}. Supposons que la précédence satisfasse les conditions du théorème 6:

$*>+$

si k et k' sont dans F_{AC}, $k>_F k'$ => k=* et k'=+

si k est dans F_{AC} et $k\neq *$, alors k est minimal dans F

si f est dans F, $* >_F f$ => f=+.

Lemme 11 : Si σx est de la forme $k(u_1,\ldots u_p)$, où $k \neq +,*$ et $u_1,\ldots u_p$ sont des termes clos dans T(F,X), alors s>t => σs>σt.

Preuve

D'après les lemmes 7 et 8, $[\sigma s\downarrow] = [[\sigma\downarrow]\ [s\downarrow]\downarrow]$. Puisque $k \neq +,*$, le terme $[\sigma\downarrow]\ [s\downarrow]$ est irréductible por les règles d'aplatissement et de distributivite. Donc, $[[\sigma\downarrow]\ [s\downarrow]\downarrow] = [\sigma\downarrow]\ [s\downarrow]$. De la même façon, $[\sigma t\downarrow] = [\sigma\downarrow]\ [t\downarrow]$. Par hypothèse, $[s\downarrow] >_{RPO} [t\downarrow]$. La stabilité par instanciation du RPO permet de conclure. □

3. Les cas litigieux

Introduisons alors la notion d'occurence mère, qui sera un concept décisif dans la preuve du lemme suivant. La forme D-irréductible aplatie $[\sigma s\downarrow]$ du terme $[\sigma\downarrow]\ [s\downarrow]$ dépend en effet de l'occurence sous laquelle on instancie. Si à cette occurence on a le symbole +, et si le symbole de tête de $[\sigma\downarrow]$ est +, on aplatira $[\sigma\downarrow]\ [s\downarrow]$ suivant +. De même, si à cette occurence on a *, et si le symbole de tête de $[\sigma\downarrow]$ est *. Si par contre, à l'occurence mère de la variable instanciée, on a *, et si le symbole de tête de $[\sigma\downarrow]$ est +, on appliquera la distributivité au terme $[\sigma\downarrow][s\downarrow]$. Illustrons ce mécanisme avec quelques exemples.

Exemple 8 : Soit $[s\downarrow] = x+y$. A l'occurence mère des deux variables se trouve le symbole +. Soit une substitution close élementaire σ telle que $[\sigma x\downarrow] = a+b$, a et b étant des constantes de F. Le symbole de tête de la substitution est +. Donc le terme instancié $[\sigma\downarrow][s\downarrow] = (a+b)+y$ s'aplatit suivant + pour donner $[\sigma s\downarrow] = a+b+y$.

Exemple 9 : Soit alors $[s\downarrow] = f((x*y),z)$. A l'occurence mère de x, on a le symbole *. Soit une substitution σ telle que $[\sigma x\downarrow] = a+b$. Le symbole de tête de la substitution est +. Donc la règle de distributivité droite de + par rapport à * s'applique au terme instancié $[\sigma\downarrow][s\downarrow] = f(((a+b)*y),z)$ pour donner $[\sigma s\downarrow] = f(((a*y)+(b*y)),z)$.

Définition 24 : L'occurence mère d'une variable dans un terme est l'occurence préfixe stricte de cette variable. Soit $i_1 \ldots i_n \in N^n$ l'occurence d'une variable dans un terme. Son occurence mère est l'occurence $i_1 \ldots i_{n-1}$.

Lemme 12 : Si σx est de la forme $u_1 * u_2$, où u_1 et u_2 sont des termes clos de T(F,X), alors $s > t \Rightarrow \sigma s > \sigma t$.

Preuve

Remarquons tout d'abord que si σx est de la forme $u_1 * u_2$, alors $[\sigma x\downarrow]$ peut être soit de la forme $[u_1\downarrow]*\ldots*[u_p\downarrow]$ soit de la forme $[u_1\downarrow]+\ldots+[u_p\downarrow]$, où $u_1, \ldots u_p$ sont des termes de T(F,X). Notons $[s\downarrow] = f([s_1\downarrow],\ldots[s_m\downarrow])$ et $[t\downarrow] = g([t_1\downarrow],\ldots[t_n\downarrow])$. On raisonnera par induction structurelle sur T(F,X), en distinguant les deux formes possibles de $[\sigma x\downarrow]$.

A. $[\sigma x\downarrow]$ est de la forme $[u_1\downarrow]*\ldots*[u_p\downarrow]$

Distinguons alors les trois cas de la définition du RPO pouvant apparaître dans l'hypothèse du lemme, soit $[s\downarrow] >_{RPO} [t\downarrow]$.

$\underline{f=g}$ $\underline{et}$ $\{[s_1\downarrow],\ldots[s_m\downarrow]\} >>_{RPO} \{[t_1\downarrow],\ldots[t_n\downarrow]\}$

Si $[s_i\downarrow]\neq x$ pour tout i, $[t_j\downarrow]\neq x$ pour tout j, ou $f\neq *$, alors $[\sigma s\downarrow] = f([\sigma s_1\downarrow],\ldots[\sigma s_m\downarrow])$ et $[\sigma t\downarrow] = f([\sigma t_1\downarrow],\ldots[\sigma t_n\downarrow])$. En effet, les symboles de tête de $[s_i\downarrow]$ et $[t_j\downarrow]$ ne sont ni +, ni *. Par hypothèse d'induction et par définition de $>>_{RPO}$, $\{[\sigma s_1\downarrow],\ldots[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_1\downarrow],\ldots[\sigma t_n\downarrow]\}$. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$.

Si $f=*$ et s'il existe i tel que $[s_i\downarrow]=x$, et $[t_j\downarrow]\neq x$ pour tout j, alors $[\sigma s\downarrow] = *([\sigma s_1\downarrow],\ldots[u_1\downarrow],..[u_p\downarrow],\ldots[\sigma s_m\downarrow])$ où $[\sigma s_i\downarrow] = [u_1\downarrow]*\ldots*[u_p\downarrow]$. D'autre part, $[\sigma t\downarrow] = *([\sigma t_1\downarrow],\ldots[\sigma t_n\downarrow])$. On a $\{[s_1\downarrow],..[s_{i-1}\downarrow],[s_{i+1}\downarrow],\ldots[s_m\downarrow]\} >>_{RPO} \{[t_1\downarrow],\ldots[t_n\downarrow]\}$ car x ne majore aucun des $[t_j\downarrow]$. Par hypothèse d'induction et par définition de $>>_{RPO}$, $\{[\sigma s_1\downarrow],..[\sigma s_{i-1}\downarrow],[\sigma s_{i+1}\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_1\downarrow],\ldots[\sigma t_n\downarrow]\}$. Donc $\{[s_1\downarrow],..[u_1\downarrow],..[u_p\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_1\downarrow],\ldots[\sigma t_n\downarrow]\}$. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$.

Si $f=*$, et s'il existe j tel que $[t_j\downarrow]=x$, et $[s_i\downarrow]\neq x$ pour tout i, alors $[\sigma t\downarrow] = [\sigma t_1\downarrow]*..*[u_1\downarrow]*..*[u_p\downarrow]..*[\sigma t_n\downarrow]$. Par hypothèse d'induction et par définition de $>>_{RPO}$, $\{[\sigma s_1\downarrow],\ldots[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_1\downarrow],..[\sigma x\downarrow],..[\sigma t_n\downarrow]\}$. Donc $\{[\sigma s_1\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_i\downarrow],..[u_i\downarrow],..[u_p\downarrow],..[\sigma t_n\downarrow]\}$ car $[\sigma x\downarrow] >_{RPO} [u_1\downarrow],\ldots[u_p\downarrow]$ par propriété sous-terme du RPO.

Si $f=*$, et s'il existe i tel que $[s_i\downarrow]=x$, et s'il existe j tel que $[t_j\downarrow]=x$, alors $\{[\sigma s_1\downarrow],..[\sigma s_{i-1}\downarrow],[\sigma s_{i+1}\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO}$

$\{[\sigma t_1\downarrow],..[\sigma t_{j-1}\downarrow],[\sigma t_{j+1}\downarrow],..[\sigma t_n\downarrow]\}$. Donc

$\{[\sigma s_1\downarrow],..[u_1\downarrow],..[u_p\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_1\downarrow],..[u_1\downarrow],..[u_p\downarrow],..[\sigma t_n\downarrow]\}$.

$f>g$ et $[s\downarrow] >_{RPO} [t_j\downarrow]$ pour tout j

Si $g=*$ et s'il existe j tel que $[t_j\downarrow]=x$, alors $[\sigma s\downarrow] = f([\sigma s_1\downarrow],...[\sigma s_m\downarrow])$, et $[\sigma t\downarrow] = [\sigma t_1\downarrow]*..*[u_1\downarrow]..*[u_p\downarrow]..*[\sigma t_n\downarrow]$. Remarquons que, par hypothèse d'induction, $[\sigma s\downarrow] >_{RPO} [\sigma t_k\downarrow]$ pour tout $k\neq j$. D'autre part, $[\sigma s\downarrow] >_{RPO} [\sigma x\downarrow]=[u_1\downarrow]*...*[u_p\downarrow]$. Donc $[\sigma s\downarrow] >_{RPO} [u_1\downarrow],...[u_p\downarrow]$ par propriété sous-terme du RPO. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$.

Sinon, $[\sigma t\downarrow] = [\sigma t_1\downarrow]*...*[\sigma t_n\downarrow]$. Par hypothèse d'induction, $[\sigma s\downarrow] >_{RPO} [\sigma t_j\downarrow]$ pour tout j. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$.

Il existe i tel que $[s_i\downarrow] \geq_{RPO} [t\downarrow]$

Si $f=*$ et $[s_i\downarrow]=x$, alors $[\sigma s\downarrow] = [\sigma s_1\downarrow]*..*[u_1\downarrow]..*[u_p\downarrow]..*[\sigma s_m\downarrow]$. Dans ce cas, $[t\downarrow]=x$ alors $[\sigma t\downarrow] = [u_1\downarrow]*...*[u_p\downarrow]$. L'inéquation suivante donne le résultat: $\{[\sigma s_1\downarrow],..[u_1\downarrow],..[u_p\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[u_1\downarrow],...[u_p\downarrow]\}$.

Sinon, $[\sigma s\downarrow] = f([\sigma s_1\downarrow],...[\sigma s_m\downarrow]\downarrow]$. Par hypothèse d'induction, $[\sigma s_i\downarrow] \geq_{RPO} [\sigma t\downarrow]$. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$.

B. $[\sigma x\downarrow]$ est de la forme $[u_1\downarrow]+...+[u_p\downarrow]$

S'il n'existe ni dans s ni dans t une occurence mère o de x, telle que $s(o)=*$ ou $t(o)=*$, alors la preuve se fait comme dans le cas précédent A, en remplaçant $*$ par $+$.

S'il existe dans s ou t une occurence mère o de x telle que $s(o)=*$ ou

$t(o)=*$, alors la distributivité s'applique, lors du calcul de $[\sigma s\downarrow]$ à partir de $[\sigma\downarrow][s\downarrow]$. Comme dans A, distinguons les trois cas de la définition du RPO qui peuvent apparaître dans l'hypothèse du lemme, soit $[s\downarrow] >_{RPO} [t\downarrow]$.

1. f=g et $\{[s_1\downarrow],\ldots[s_m\downarrow]\} >>_{RPO} \{[t_1\downarrow],\ldots[t_n\downarrow]\}$

Si $f\neq *$, ou $f=*$ et $[s_i\downarrow]\neq x$ pour tout i et $[t_j\downarrow]\neq x$ pour tout j, alors $[\sigma s\downarrow] = [f([\sigma s_1\downarrow],\ldots[\sigma s_m\downarrow])]_+$, et $[\sigma t\downarrow] = [f([\sigma t_1\downarrow],\ldots[\sigma t_n\downarrow])]_+$. Par hypothèse d'induction, $\{[\sigma s_1\downarrow],\ldots[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_1\downarrow],\ldots[t_n\downarrow]\}$. Donc $f([s_1\downarrow],\ldots[\sigma s_m\downarrow]) >_{RPO} f([\sigma t_1\downarrow],\ldots[\sigma t_n\downarrow])$. On conclut avec le lemme 3.

Notons $+(u_k)[k \text{ dans } [1,p]]$ pour le terme $+(u_1,\ldots u_p)$.

Si $f=*$ et s'il existe i tel que $[s_i\downarrow]=x$ (on suppose ici qu'un tel i est unique mais le raisonnement se généralise facilement à plusieurs) et $[t_j\downarrow]\neq x$ pour tout j, alors $[\sigma t\downarrow] = [\sigma t_1\downarrow]*\ldots*[\sigma t_n\downarrow]$. D'autre part, $[\sigma s\downarrow] = [[\sigma s_1\downarrow]*..*([u_1\downarrow]+\ldots+[u_p\downarrow])*..*[\sigma s_m\downarrow]\downarrow] = +([[u_k\downarrow]*[\sigma s_1\downarrow]*..*[\sigma s_{i-1}\downarrow]*[\sigma s_{i+1}\downarrow]*..*[\sigma s_m\downarrow]]_*)\ [k \text{ dans } [1,p]]$ de la même façon que dans le lemme 5. Remarquons alors que pour tout k dans [1,m], le symbole de tête de $[\sigma s_k\downarrow]$ ne peut être $*$. Donc pour tout k dans [1,p], $[[u_k\downarrow],[\sigma s_1\downarrow],\ldots[\sigma s_m\downarrow]]_* = [u_{k1}\downarrow]*..*[u_{kh}\downarrow]*[\sigma s_1\downarrow]*..*[\sigma s_m\downarrow]$. Par hypothèse, $\{[s_1\downarrow],..[s_{i-1}\downarrow],[s_{i+1}\downarrow],..[s_m\downarrow]\} >>_{RPO} \{[t_1\downarrow],\ldots[t_n\downarrow]\}$ car $[s_i\downarrow]=x$. Donc par hypothèse d'induction, $\{[\sigma s_1\downarrow],..[\sigma s_{i-1}\downarrow],[\sigma s_{i+1}\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_1\downarrow],\ldots[\sigma t_n\downarrow]\}$. Donc pour tout k dans [1,p], $\{[\sigma s_1\downarrow],..[u_{k1}\downarrow],..[u_{kh}\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_1\downarrow],\ldots[\sigma t_n\downarrow]\}$. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$.

Si $f=*$ et $[s_i\downarrow]\neq x$ pour tout i et s'il existe j tel que $[t_j\downarrow]=x$,

alors, de la même façon que précedemment, $[\sigma s\downarrow] = [\sigma s_1\downarrow]*\ldots*[\sigma s_m\downarrow]$, et $[\sigma t\downarrow] = +([[u_k\downarrow]*[\sigma t_1\downarrow]*..[\sigma t_{j-1}\downarrow]*[\sigma t_{j+1}\downarrow]*..[\sigma t_n\downarrow]]_*)$ [k dans [1,p]]. Par hypothèse d'induction et par définition de $>>_{RPO}$, $\{[\sigma s_1\downarrow],\ldots[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_1\downarrow],..[\sigma t_j\downarrow],..[\sigma t_n\downarrow]\}$. Donc $\{[\sigma s_1\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_1\downarrow],..[u_{k1}\downarrow],..[u_{kh}\downarrow],..[t_n\downarrow]\}$ pour tout k dans [1,p], par propriété sous-terme du RPO. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$.

Si f=* et s'il existe i tel que $[s_i\downarrow]=x$ et s'il existe j tel que $[t_j\downarrow]=x$, alors $[\sigma s\downarrow] = +([\sigma s_1\downarrow]*..[\sigma s_{i-1}\downarrow]*[u_{k1}\downarrow]*..[u_{kh}\downarrow]*[\sigma s_{i+1}\downarrow]*..[\sigma s_m\downarrow])$ [k dans [1,p]]. De même, $[\sigma t\downarrow] = +([\sigma t_1\downarrow]*..[\sigma t_{j-1}\downarrow]*[u_{k1}\downarrow]*..[u_{kh}\downarrow]*[\sigma t_{j+1}\downarrow]*..[\sigma t_n\downarrow])$ [k dans [1,p]]. Par hypothèse d'induction et définition de $>>_{RPO}$, $\{[\sigma s_1\downarrow],..[\sigma s_{i-1}\downarrow],[\sigma s_{i+1}\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_1\downarrow],..[\sigma t_{j-1}\downarrow],[\sigma t_{j+1}\downarrow],..[\sigma t_n\downarrow]\}$. Donc $[\sigma s_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[s_m\downarrow] >_{RPO} [\sigma t_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[t_n\downarrow]$ pour tout k dans [1,p]. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$.

2. f>g et $[s\downarrow] >_{RPO} [t_j\downarrow]$ pour tout j

Remarquons tout d'abord que les conditions sur la précédence nous interdisent d'avoir f=* avec g≠+. Distinguons alors plusieurs cas selon les valeurs de f et de g.

2.1. f=* et g=+

Si $[s_i\downarrow]\neq x$ pour tout i et $[t_j\downarrow]\neq x$ pour tout j, alors $[\sigma s\downarrow] = [\sigma s_1\downarrow]*\ldots*[\sigma s_m\downarrow]$, $[\sigma t\downarrow] = [[\sigma t_1\downarrow]+\ldots+[\sigma t_n\downarrow]]_+$. Par hypothèse d'induction, $[\sigma s\downarrow] >_{RPO} [\sigma t_j\downarrow]$ pour tout j. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t_1\downarrow]+\ldots+[\sigma t_n\downarrow]$. On conclut avec le lemme 3.

S'il existe i tel que $[s_i\downarrow]=x$ et $[t_j\downarrow]\neq x$ pour tout j, alors $[\sigma s\downarrow]$ = $+([[\sigma s_1\downarrow]*..[\sigma s_{i-1}\downarrow]*[u_k\downarrow]*[\sigma s_{i+1}\downarrow]..[\sigma s_m\downarrow]]_*)$ [k dans [1,p]], soit $+([\sigma s_1\downarrow]*..*[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma s_m\downarrow])$ [k dans [1,p]] si on aplatit le terme suivant le symbole *. D'autre part, $[\sigma t\downarrow] = [[\sigma t_1\downarrow]+...+[\sigma t_n\downarrow]]_+$. Par hypothèse, $[s_1\downarrow]*...*[s_m\downarrow] >_{RPO} [t_j\downarrow]$ pour tout j de [1,n]. Deux cas se présentent suivant que le symbole de tête de $[t_j\downarrow]$ est * ou non.

- $[t_j\downarrow]$ est de la forme $[t_{j1}\downarrow]*...*[t_{jq}\downarrow]$

D'après l'hypothèse et par définition du RPO, $\{[s_1\downarrow],...[s_m\downarrow]\} >>_{RPO} \{[t_{j1}\downarrow],...[t_{jq}\downarrow]\}$. Nommons cette inégalité (I) et distinguons encore deux sous-cas.

S'il existe a tel que $[t_{ja}\downarrow]=x$, alors $[\sigma t_j\downarrow]= +[v_k]$ [k dans [1,p]], où les v_k sont de la forme $[\sigma t_{j1}\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma t_{jq}\downarrow]$. Le terme $[\sigma t\downarrow]$ s'aplatira donc suivant $[\sigma t_j\downarrow]$ en $[[\sigma t_1\downarrow]+..v_1+..v_p+..[\sigma t_n\downarrow]]$. Dans ce cas, il suffit donc de montrer que $[\sigma s_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma s_m\downarrow] >_{RPO} v_k$ pour tout k de [1,p]. D'après l'inégalité multiensemble (I), on obtient $\{[s_1\downarrow],..[s_{i-1}\downarrow],[s_{i+1}\downarrow],..[s_m\downarrow]\} >>_{RPO} \{[t_{j1}\downarrow],..[t_{ja-1}\downarrow],[t_{ja+1}\downarrow],..[t_{jq}\downarrow]\}$. Par hypothèse d'induction, $\{[\sigma s_1\downarrow],..[\sigma s_{i-1}\downarrow],[\sigma s_{i+1}\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_{j1}\downarrow],..[\sigma t_{ja-1}\downarrow],[\sigma t_{ja+1}\downarrow],..[\sigma t_{jq}\downarrow]\}$. Donc $\{[\sigma s_1\downarrow],..[\sigma s_{i-1}\downarrow],[u_{k1}\downarrow],..[u_{kh}\downarrow],[\sigma s_{i+1}\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_{j1}\downarrow],..[\sigma t_{ja-1}\downarrow],[u_{k1}\downarrow],..[u_{kh}\downarrow],[\sigma t_{ja+1}\downarrow],..[\sigma t_{jq}\downarrow]\}$. Donc $[\sigma s_1\downarrow]*..*[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma s_m\downarrow] >_{RPO} v_k$ pour tout k de [1,p].

S'il n'existe pas a tel que $t_{ja}=x$, alors $[\sigma t_j\downarrow] = [\sigma t_{j1}\downarrow]*...*[\sigma t_{jq}\downarrow]$. De l'inégalité multiensemble (I), on déduit $\{[s_1\downarrow],..[s_{i-1}\downarrow],[s_{i+1}\downarrow],..[s_m\downarrow]\} >>_{RPO} [[t_{j1}\downarrow],...[t_{jq}\downarrow]]$ car x ne majore

aucun des t_{ja}. Par hypothèse d'induction, $\{[\sigma s_1\downarrow],..[\sigma s_{i-1}\downarrow],[\sigma s_{i+1}\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_{j1}\downarrow],...[\sigma t_{jq}\downarrow]\}$. A fortiori, $\{[\sigma s_1\downarrow],..[\sigma s_{i-1}\downarrow],[u_{k1}\downarrow],..[u_{kh}\downarrow],[\sigma s_{i+1}\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[\sigma t_{j1}\downarrow],...[\sigma t_{jq}\downarrow]\}$. Donc $[\sigma s_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma s_m\downarrow] >_{RPO} [\sigma t_j\downarrow]$.

- $[t_j\downarrow]$ n'est pas de la forme $[t_{j1}\downarrow]*...*[t_{jq}\downarrow]$

Dans ce cas, il existe a tel que $[s_a\downarrow] \geqq_{RPO} [t_j\downarrow]$ et $a\neq i$ car sinon, on aurait $x \geqq_{RPO} [t_j\downarrow]$, donc $[t_j\downarrow]=x$, ce qui contredirait l'hypothèse. Par hypothèse d'induction, $[\sigma s_a\downarrow] \geqq_{RPO} [\sigma t_j\downarrow]$, donc par définition du RPO, $[\sigma s_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma s_m\downarrow] >_{RPO} [\sigma t_j\downarrow]$. Donc, au total, on obtient $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$.

Si $[s_i\downarrow]\neq x$ pour tout i et s'il existe j tel que $[t_j\downarrow]=x$, alors $[\sigma s\downarrow] = [\sigma s_1\downarrow]*...*[\sigma s_m\downarrow]$ et $[\sigma t\downarrow] = [[\sigma t_1\downarrow]+..[\sigma t_{j-1}\downarrow]+[u_1\downarrow]+..[u_p\downarrow]+[\sigma t_{j+1}\downarrow]+..[\sigma t_n\downarrow]]_+$. Par hypothèse d'induction, $[\sigma s\downarrow] >_{RPO} [\sigma t_k\downarrow]$ pour tout k. Donc $[\sigma s\downarrow] >_{RPO} [u_1\downarrow],...[u_p\downarrow]$ par propriété sous-terme du RPO car $[\sigma t_j\downarrow] = [u_1\downarrow]+...+[u_p\downarrow]$. On conclut avec le lemme 3.

S'il existe i tel que $[s_i\downarrow]=x$ et s'il existe j tel que $[t_j\downarrow]=x$, alors $[\sigma s\downarrow] = +([\sigma s_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma s_m\downarrow])$ [k dans [1,p]] et $[t\downarrow]$ est comme dans le cas précédent. Ce cas se traite exactement comme le cas où il existe i tel que $[s_i\downarrow]=x$ et $[t_j\downarrow]\neq x$ pour tout j, à la différence près que dans $[\sigma t\downarrow]$, le sous-terme $[\sigma t_j\downarrow]$ est remplacé par les sous-termes $[u_1\downarrow],...[u_p\downarrow]$. Il reste donc à montrer que tout u_r, r variant entre 1 et p, est majoré par un des $[\sigma s_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma s_m\downarrow]$, k variant. Ceci est vrai si on prend k=r, car $[u_r\downarrow] = [u_{r1}\downarrow]*...*[u_{rh}\downarrow]$, si le symbole racine de $[u_r\downarrow]$ est $*$. Si le symbole racine de $[u_r\downarrow]$ n'est pas $*$, le terme

$[\sigma s_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma s_m\downarrow]$ se réduit à $[\sigma s_1\downarrow]*..[u_k\downarrow]*..[\sigma s_m\downarrow]$, qui majore $[u_k\downarrow]$.

2.2. $f\neq *$ et $g=+$

Si $[t_j\downarrow]\neq x$ pour tout j, la preuve se fait comme dans le premier cas de 2.1, en remplaçant $*$ par f.

S'il existe j tel que $[t_j\downarrow]=x$, alors $[\sigma t\downarrow] = [[\sigma t_1\downarrow]+..[u_1\downarrow]+..[u_p\downarrow]+..[\sigma t_n\downarrow]]_+$. Par hypothèse d'induction, $[\sigma s\downarrow] >_{RPO} [\sigma t_j\downarrow]$ pour tout j. Par propriété sous-terme du RPO, on obtient $[\sigma s\downarrow] >_{RPO} [u_1\downarrow],...[u_p\downarrow]$. On conclut avec le lemme 3.

2.3. $f\neq *$ et $g=*$

Si $[t_j\downarrow]\neq x$ pour tout j, alors $[\sigma t\downarrow] = [\sigma t_1\downarrow]*...*[\sigma t_n\downarrow]$, et $[\sigma s\downarrow] = f([\sigma s_1\downarrow],...,[\sigma s_m\downarrow]\downarrow]$. Par hypothèse d'induction, $[\sigma s\downarrow] >_{RPO} [\sigma t_j\downarrow]$ pour tout j. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$.

S'il existe j tel que $[t_j\downarrow]=x$, alors $[\sigma t\downarrow] = +([\sigma t_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma t_n\downarrow])$ [k dans [1,p]]. Par hypothèse d'induction, $[\sigma s\downarrow] >_{RPO} [\sigma t_k\downarrow]$ pour tout k. Donc $[\sigma s\downarrow] >_{RPO} [u_{k1}\downarrow],...[u_{kh}\downarrow]$ pour tout k dans [1,p]. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma t_n\downarrow]$ pour tout k. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$ car $f>+$.

Si $f\neq *$ et $g\neq *,+$, la preuve se fait comme dans le premier cas de 2.3.

3. Il existe i tel que $[s_i\downarrow] \geq_{RPO} [t\downarrow]$

Par hypothèse d'induction, $[\sigma s_i\downarrow] >_{RPO} [\sigma t\downarrow]$. Distinguons le cas où

$[s_i\downarrow]=x$.

3.1. $[s_i\downarrow]=\underline{x}$ (donc $[\underline{t}\downarrow]=\underline{x}$)

$\underline{f}=*$

Dans ce cas, $[\sigma s\downarrow] = +([\sigma s_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma s_m\downarrow])$ [k dans [1,p]], et $[\sigma t\downarrow] = [u_1\downarrow]+...+[u_p\downarrow]$. Pour tout k dans [1,p], on a $[\sigma s_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma s_m\downarrow] >_{RPO} [u_k\downarrow]=[u_{k1}\downarrow]*...*[u_{kh}\downarrow]$. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$.

$\underline{f}=+$

Dans ce cas, $[\sigma s\downarrow] = [[\sigma s_1\downarrow]+..[u_1\downarrow]+..[u_p\downarrow]+..[\sigma s_m\downarrow]]_+$. Il est clair que $\{[\sigma s_1\downarrow],..[u_1\downarrow],..[u_p\downarrow],..[\sigma s_m\downarrow]\} >>_{RPO} \{[u_1\downarrow],...[u_p\downarrow]\}$. On conclut en utilisant la définition du RPO et le lemme 3.

$\underline{f}\neq+,*$

Le résultat est évident par propriété sous-terme du RPO car $[\sigma s\downarrow] = f([\sigma s_1\downarrow],..[\sigma x\downarrow],..[\sigma s_m\downarrow])$ et $[\sigma t\downarrow] = [\sigma x\downarrow]$.

3.2. $[s_i\downarrow]\neq\underline{x}$

Si f=* et s'il existe j tel que $[s_j\downarrow]=x$, alors $[\sigma s\downarrow] = +([\sigma s_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma s_m\downarrow])$ [k dans [1,p]]. Par hypothèse d'induction, $[\sigma s_i\downarrow] \geqq_{RPO} [\sigma t\downarrow]$. Donc $[\sigma s_1\downarrow]*..[u_{k1}\downarrow]*..[u_{kh}\downarrow]*..[\sigma s_m\downarrow] >_{RPO} [\sigma t\downarrow]$ pour tout k. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$.

Si f=+ et s'il existe j tel que $[s_j\downarrow]=x$, alors $[\sigma s\downarrow] = [[\sigma s_1\downarrow]+..[\sigma s_i\downarrow]+..[u_1\downarrow]+..[u_p\downarrow]+..[\sigma s_m\downarrow]]_+$. Puisque $[\sigma s_i\downarrow] \geqq_{RPO} [\sigma t\downarrow]$, on a $[\sigma s_1\downarrow]+...+[\sigma s_i\downarrow]+...+[\sigma s_m\downarrow] >_{RPO} [\sigma t\downarrow]$. On conclut avec le lemme 3.

Si $f \neq +,*$ ou $[s_j\downarrow] \neq x$ pour tout j, alors $[\sigma s\downarrow] = f([\sigma s_1\downarrow],\ldots[\sigma s_m\downarrow])$. Donc $[\sigma s\downarrow] >_{RPO} [\sigma t\downarrow]$ car $[\sigma s_i\downarrow] \geq_{RPO} [\sigma t\downarrow]$. □

Lemme 13 : Si σx est de la forme u_1+u_2, où u_1 et u_2 sont des termes clos dans T(F,X), alors $s>t \Rightarrow \sigma s > \sigma t$.

Preuve

La preuve se fait comme dans le cas B du lemme précédent, car $[\sigma x\downarrow]$ est de la forme $[u_1\downarrow]+\ldots+[u_p\downarrow]$. □

Enonçons alors le théorème de stabilité par instanciation sur les termes clos.

Théorème 7 : $s>t \Rightarrow \sigma s > \sigma t$ pour toute substitution close σ.

Preuve

La preuve découle immédiatement des lemmes 9, 10, 11, 12, et 13. □

Dans le cas où l'ensemble C des constantes n'est pas vide, le théorème précédent équivaut au théorème général de stabilité par instanciation sur les substitutions quelconques.

Théorème 8 : $s>t \Rightarrow \sigma s > \sigma t$ pour toute substitution σ.

Preuve

S'il existe au moins une constante dans l'algèbre des termes, alors le problème de la terminaison sur les termes clos de T(F,X) équivaut au problème de la terminaison sur les termes quelconques de T(F,X). En effet, s'il n'existe pas de chaîne infinie de termes quelconques, il n'existe pas

de chaîne infinie de termes clos. Réciproquement et par l'absurde, supposons qu'il existe une chaîne infinie de termes quelconques. Comme il existe au moins une constante dans T(F,X), il existe une substitution close s'appliquant à chaque terme non clos de la chaîne précédente. D'où l'obtention d'une chaîne infinie de termes clos, ce qui contredit l'hypothèse. □

Nous sommes alors en mesure de citer le théorème général de terminaison qui constitue la clé de notre travail.

Théorème 9 : Un système de réécriture R termine modulo la théorie associative-commutative si pour toute règle $g \rightarrow d$ de R, on a: $g >_{NFLO} d$.

CHAPITRE 6: EXEMPLES

Montrons alors le champ d'application du NFLO sur une série d'exemples usuels. Ces exemples sont pour la plupart traités par REVE3, qui, utilisant pour le moment un ordre non AC-commutant, ne garantit pas la terminaison AC.

Exemple 1

Considérons tout d'abord l'axiomatisation suivante des groupes abéliens:

$$e*x \rightarrow x$$

$$i(x)*x \rightarrow e$$

où $*$ est un symbole AC. Tous les termes de ce système sont déjà en forme irréductible aplatie. On a $e*x >_{RPO} x$ par propriété sous-terme du RPO,et $i(x)*x >_{RPO} e$ avec la précédence $i >_F e$. Donc le système est AC-noethérien.

Exemple 2

Soit la définition d'un anneau commutatif:

$$0+x \rightarrow x$$

$$-x+x \rightarrow 0$$

$$(x+y)*z \rightarrow (x*z)+(y*z)$$

$$x*(y+z) \rightarrow (x*y)+(x*z)$$

où $+$ et $*$ sont des symboles AC. La première équation est orientée par propriété sous-terme du RPO. Pour la deuxième, si on pose $\_ >_F 0$, alors

$-x+x >_{RPO} 0$. Pour la troisième équation, qui est la règle de distributivité elle-même, on a $(x+y)*z > (x*z)+(y*z)$ par définition de l'ordre NFLO. La quatrième équation se traite comme la troisième.

Exemple 3

Considérons la définition suivante d'un anneau commutatif unitaire:

$$0+x \rightarrow x$$

$$-x+x \rightarrow 0$$

$$(x+y)*z \rightarrow (x*z)+(y*z)$$

$$x*(y+z) \rightarrow (x*y)+(x*z)$$

$$x*1 \rightarrow x$$

$$1*x \rightarrow x$$

où + et * sont associatifs-commutatifs. Par propriété sous-terme du RPO, $0+x >_{RPO} x$. Si $\_>_F 0$, on obtient $-x+x >_{RPO} 0$. Comme dans l'exemple précédent, $(x+y)*z >_{RPO} (x*z)+(y*z)$ et $x*(y+z) >_{RPO} (x*z)+(y*z)$. Pour terminer, $x*1 >_{RPO} x$ et $1*x >_{RPO} x$ par propriété sous-terme du RPO.

Exemple 4

Donnons une axiomatisation du treillis distributif libre:

$$(y+z)*x \rightarrow (x*y)+(x*z)$$

$$(x*y)+x \rightarrow x$$

$$(x+y)*x \rightarrow x$$

où + et * satisfont les axiomes d'associativité et de commutativité. Pour la première règle, $[(y+z)*x\downarrow] = (y*x)+(z*x) \simeq (x*y)+(x*z)$. Or $(y+z)*x =_{AC} x*(y+z) \rightarrow_D (x*y)+(x*z)$, soit $(y+z)*x \rightarrow_{D/AC} (x*y)+(x*z)$. Donc

$(y+z)*x > (x*y)+(x*z)$. La deuxième règle est orientée grâce à la propriété sous-terme du RPO. Pour la troisième, $[(x+y)*x\downarrow] = (x*x)+(y*x) >_{RPO} x$.

Exemple 5

L'axiomatisation suivante de la fonction puissance de 2 est un système AC-noethérien.

$$h(0) \rightarrow s(0)$$

$$h(s(0)) \rightarrow s(s(0))$$

$$h(s(0)) \rightarrow *(s(s(0)),h(0))$$

$$h(x+1) \rightarrow *(2,h(x))$$

$$h(x+y) \rightarrow h(x)*h(y)$$

où encore, + et * sont associatifs-commutatifs. En effet, tous les termes du système sont en forme irréductible aplatie. Et si on pose $h >_F s$, on a $h(0) >_{RPO} s(0)$ et $h(s(0)) >_{RPO} s(s(0))$. Avec $h >_F *$, $h(s(0)) >_{RPO} *(s(s(0)),h(0))$ car $h(s(0)) >_{RPO} s(s(0))$ et $h*s(0)) >_{RPO} h(0)$. Avec $h >_F 2$, on obtient $h(x+1) >_{RPO} *(2,h(x))$, car $h(x+1) >_{RPO} h(x)$. Sans hypothèse supplémentaire à la précédence, $h(x+y) >_{RPO} h(x)*h(y)$ car $h >_F *$. Les conditions de précédence imposées par le NFLO sont bien respectées.

Exemple 6

Soit une axiomatisation des anneaux booléens:

$$(1) \quad 0+x \rightarrow x$$

$$(2) \quad 0*x \rightarrow 0$$

$$(3) \quad 1*x \rightarrow x$$

$$(4) \quad x*x \rightarrow x$$

(5) $(x+y)*z \rightarrow (x*z)+(y*z)$

(6) $x+x \rightarrow 0$

(7) $not(x) \rightarrow 1+x$

(8) $x|y \rightarrow (x*y)+(x+y)$

(9) $x \text{ imp } y \rightarrow (x*y)+(x+1)$

(10) $x \text{ equiv } y \rightarrow (x+y)+1$

où les opérateurs +, * sont associatifs-commutatifs. La règle (6) ne peut être orientée avec l'ordre NFLO dans les conditions de précédence que nous venons de développer. En effet, x est incomparable avec 0 et $+ \text{ non}>_F 0$.

Nous suggérons alors d'étendre la précédence à des conditions de minimalité sur les variables, en posant $x >\sim 0$. Nous y adjoindrons la condition $c\sim c'$ pour toutes les constantes c et c' de F, car x peut s'instancier en c ou en c'. Nous définirons la relation $\sim$ comme étant une relation d'équivalence sur F, contenant l'égalité syntaxique. La définition du RPO, donc de l'ordre NFLO devra ainsi être adaptée et étendue au cas où on compare une variable et une constante: si $x >\sim c$ alors $x >\sim_{RPO} c$. Il semblerait que le NFLO conserve les mêmes propriétés de F-compatibilité et de stabilité par instanciation dans ce cas.

Exemple 7

Soit une spécification des entiers de Péano:

$x+0 \rightarrow x$

$x+s(y) \rightarrow s(x+y)$

$x*0 \rightarrow x$

$x*s(y) \rightarrow x+(x*y)$

où + et * sont ici encore associatifs-commutatifs. Cet exemple ne peut être, lui non plus, pris en compte par le NFLO avec la précédence que nous imposons. En effet, pour orienter la deuxième règle, il faudrait que $+ >_F s$, ce qui trahirait la condition de minimalité de l'opérateur +. Nous allons voir dans le chapitre suivant qu'une telle règle se traite avec un ordre similaire au NFLO, où la règle annexe de distributivité est remplacée par une règle de la forme $x+s(y) \rightarrow s(x+y)$. Nous allons étudier cette extension de la théorie dans le cas de l'endomorphisme.

CHAPITRE 7: UNE EXTENSION DE LA METHODE A D'AUTRES REGLES DE TRANSFORMATION

L'ordre précédemment développé, fondé sur la transformation par distributivité, ne convient pas à tout système de réécriture AC. Par exemple, considérons le système contenant la règle d'endomorphisme $f(x)+f(y)\rightarrow f(x+y)$, où f est dans F_{NAC} et + dans F_{AC}. Cette règle ne peut être orientée par l'ordre précédent >, car + est minimal pour la précédence.

Utilisons la même approche que dans le chapitre 3, en tentant de construire un ordre de réduction AC-commutant, qui puisse orienter la règle d'endomorphisme dans la direction souhaitée. Nous obtenons un ordre similaire à celui du chapitre 3, en remplaçant les règles de distributivité par le système convergent (PLUS), décrit par les deux règles suivantes:

$$f(x)+y \rightarrow f(x+y)$$

$$x+f(y) \rightarrow f(x+y).$$

La forme normale pour (PLUS) d'un terme s est notée $s\downarrow$. Comme dans le cas de la distributivité, on peut définir la forme $[s\downarrow]$ pour tout terme s de T(F,X), par $[s\downarrow]$. La définition du nouvel ordre peut être exprimée comme suit:

Définition 25 : Soient s et t deux termes de T(F,X). On a s>t si et seulement si

$[s\downarrow] >_{RPO} [t\downarrow]$ ou

$[s\downarrow] \simeq [t\downarrow]$ et $s \text{ -+}\rightarrow_{PLUS/AC} t$.

Avec le même type de contraintes sur la précédence que dans le chapitre 4 (telles que +>f), on montre que > est un ordre de réduction AC-commutant.

La preuve de F-compatibilité a été développée dans ce cas où la transformation annexe est la règle d'endomorphisme. Les arguments et schémas de preuve sont tout à fait similaires au cas de la distributivité. La validité de la F-compatibilité du RPO sur les termes aplatis provient en effet du fait qu'on "tempère" l'aplatissement par une règle qui "réduit" le terme. La distributivité et la règle d'endomorphisme ont ici le même comportement: elles font "monter" l'opérateur le plus petit vers la racine, et descendre l'opérateur le plus grand vers les feuilles du terme auquel elles sont appliquées.

De cette façon, on peut construire une famille d'ordres de réduction AC-commutants, permettant de prouver la terminaison AC de différents types de systèmes de réécriture. En effet, on peut suggérer de traiter d'autres règles de la même façon, telles que la définition de + en terme de successeur dans la spécification des entiers de Péano: $s(x)+y \rightarrow s(x+y)$; ou une propriété de * telle que $-(x)*y \rightarrow -(x*y)$.

CONCLUSION ET PERSPECTIVES

1. Bilan

Au cours de ce travail, nous nous sommes attachés à une propriété souvent méconnue et dans certains cas, encore négligée de la réécriture: le problème de la terminaison. Notre objectif a été d'approfondir un aspect de ce problème récemment abordé: la terminaison de la réécriture équationnelle, plus particulièrement dans le cas où les équations sont des axiomes d'associativité et de commutativité.

Après avoir fait état des principales méthodes utilisées pour les preuves de terminaison simple, nous avons présenté les méthodes générales de preuves proposées pour la terminaison équationnelle. Le but de ce travail était de construire un ordre adapté au cas AC. Nous nous sommes appuyés pour cela sur un théorème de Jouannaud et Muñoz exprimant que l'ordre en question devait être un ordre de réduction et posséder la propriété de AC-commutation. Nous sommes partis d'un ordre de réduction simple et connu, le RPO. Nous lui avons adjoint le processus d'aplatissement des termes à comparer pour le rendre AC-commutant, et une transformation par réécriture annexe distributive pour le rendre F-compatible. Le principe de cet ordre avait été proposé auparavant par Bachmair et Plaisted, mais les justifications théoriques n'étaient pas complètes, et la preuve de stabilité par instanciation trop restrictive.

L'apport de notre travail a donc été une justification complète des propriétés d'ordre de réduction et de AC-commutation de l'ordre précédent,

et une preuve tout à fait générale de la stabilité par instanciation. Notons aussi la nette clarification des concepts de transformation sur lesquels l'ordre est fondé. Comme nous l'avons fait remarquer au cours de ce travail, Bachmair et Plaisted intercallent les étapes de distributivité et d'aplatissement. L'application des règles de distributivité à des termes en cours d'aplatissement, donc varyadiques, nécessite l'emploi d'une infinité de règles distributives. Quant à nous, nous voyons l'opération d'aplatissement comme une interprétation des termes de T(F,X) par des termes de l'algèbre des termes aplatis FL(F,X). Cette optique permet de rapprocher cette méthode de la méthode polynomiale, où les termes de T(F,X) sont interprétés par des polynômes, et de n'utiliser qu'une distributivité, la distributivité sur les termes de T(F,X), pour laquelle l'arité des symboles AC est 2.

De par la propriété de stabilité par instanciation, l'ordre que nous proposons est implémentable et pourrait dès maintenant fournir au logiciel de réécriture équationnelle REVE3 un outil de preuve automatique de terminaison.

2. Perspectives

Sous sa forme actuelle, le NFLO ne peut toutefois considérer tous les cas de systèmes AC. Le cas de l'algèbre booléenne nécessite l'hypothèse $x>\sim 0$, qui introduit des contraintes d'ordre des variables par rapport aux constantes, et des constantes entre elles. Une extension de la précédence et de l'ordre NFLO peut donc être suggérée, dans le but de prendre en compte des règles du type $x+x \rightarrow 0$.

Il existe aussi des systèmes contenant la règle d'endomorphisme, ou tels que l'axiomatisation des entiers de Péano, pour lesquels la règle de transformation nécessaire n'est plus la distributivité mais par exemple l'endomorphisme.

Nous avons observé que toutes ces règles traduisent le même mécanisme de mouvement des opérateurs dans un terme, suivant leur poids dans la précédence, et que la preuve de F-compatibilité dans le cas de l'endomorphisme est tout à fait similaire à celle du cas distributif.

Des perspectives s'ouvrent donc dans le sens d'une généralisation de l'ordre à diverses règles annexes. Une extension de la théorie pourra se faire en plusieurs temps. Il serait tout d'abord intéressant d'abstraire les preuves développées dans ce travail sur le cas distributif, en extrayant les principes et mécanismes communs au cas distributif et aux cas similaires comme l'endomorphisme.

De plus, la structure de l'ordre que nous avons développé semble promettre de s'adapter à d'autres théories équationnelles que la théorie AC. Il serait en effet souhaitable de chercher quelles règles de réécriture annexe pourraient convenir à des théories équationnelles comme la théorie permutative.

References

1. B.Hayes, "Le probleme de Syracuse," Pour la Science, pp. 98-103, Mai 1984.

2. L. Bachmair and N. Dershowitz, "Commutation, Transformation, and Termination," Inter. Report U. of Illinois, 1985.

3. L. Bachmair and D. Plaisted, "Associative Path Orderings," in Proc. 1st Conference on Rewriting Techniques and Applications, Lecture Notes in Computer Science, vol. 202, pp. 241-254, Springer Verlag, Dijon (France), 1985.

4. F. Bellegarde, "Rewriting Systems on FP Expressions to reduce the number of Sequences Yielded," Science of Computer Programming, vol. North Holland, 1985.

5. J. A. Bergstra and J. W. Klop, "Algebra of Communicating Processes with Abstraction," Theoretical Computer Science, vol. 37, pp. 77-121, North-Holland, 1985.

6. A. Ben Cherifa and P. Lescanne, "A method for proving termination of rewriting systems, based on elementary computations on polynomials," Internal Report CRIN, Nancy, 1985.

7. C. Choppy and C. Johnen, "Petrireve: Proving Petri Net Properties With Rewriting Systems," in Proc. 1st Conference on Rewriting Techniques and Applications, Lecture Notes in Computer Science, vol. 202, pp. 271-286, Springer Verlag, Dijon (France), 1985.

8. A. Church, "The Calculi of Lambda-Conversion," Princeton U. Press, Princeton N.J., 1941.

9. S.S. Cosmadakis and P.C. Kanellakis, "Two Applications of Equational Theories to Data base Theory," in Proc. 1rst International Conference on Rewriting Techniques and Applications, Lecture Notes in Computer Science, vol. 202, pp. 107-123, Springer Verlag, Dijon (France), 1985.

10. Nachum Dershowitz, "Orderings for Term-Rewriting Systems," Theoretical Computer Science, vol. 17, pp. 279-301, 1982.

11. N. Dershowitz, J. Hsiang, N.A. Josephson, and D.A. Plaisted, "Associative-Commutative rewriting," Proceedings of the 8th IJCAI, pp. 940-944, Karlsruhe (West Germany), 1983.

12. N. Dershowitz, "Termination," in Proc. 1rst Conf. Rewriting Techniques and Applications, Lecture Notes in Computer Science, vol. 202, pp. 180-224, Springer Verlag, Dijon (France), May 1985.

13. A.J.J. Dick, "ERIL - Equational reasoning: an interactive laboratory," Proceedings of the EUROCAL Conference, Linz (Austria), 1985.

14. T. Evans, "The Word Problem for Abstract Algebras," J. London Math. Soc., vol. 26, pp. 64-71, 1951.

15. F. Fages, "Le systeme KB : manuel de reference : presentation et bibliographie, mise en oeuvre," R.G. 10.84, Greco de Programmation, Bordeaux, 1984.

16. R. Forgaard and D. Detlefs, "An incremental algorithm for proving termination of term rewriting sytems," in Proc. 1rst International

Conference on Rewriting Techniques and Applications., Lecture Notes in Computer Science, vol. 202, Springer Verlag, Dijon (France), 1985.

17. I. Gnaedig, "Trois extensions du processus d'orientation des règles de réécriture dans REVE," 83-R-097, CRIN, Nancy, 1983.

18. G. Huet and D.S. Lankford, "On the Uniform Halting Problem for Term Rewriting Systems," Rapport Laboria 283, Iria, Mars 1978.

19. G. Huet and D. Oppen, "Equations and Rewrite Rules: A Survey," in Formal Languages: Perspectives And Open Problems, ed. Book R., Academic Press, 1980.

20. G. Huet, "Confluent reductions: abstract properties and applications to term rewriting systems," J. of ACM, vol. 27, no. 4, pp. 797-821, Oct. 1980.

21. J.P. Jouannaud, P. Lescanne, and F. Reinig, "Recursive Decomposition Ordering," in Formal Description of Programming Concepts 2, ed. Bjorner D., pp. 331-346, North Holland, Garmish Partenkirschen, RFA, 1982.

22. J.P. Jouannaud, "Church-Rosser computations with equational term rewriting systems," to appear in J. ACM, 1983.

23. J.P. Jouannaud and H. Kirchner, "Completion of a set of rules modulo a set of equations," Proceedings 11th ACM Conference of Principles of Programming Languages, Salt Lake City (Utah, USA), 1984.

24. J.P. Jouannaud and M. Munoz, "Termination of a set of rules modulo a set of equations," Proceedings 7th Conference on Automated Deduction,

vol. 170, Napa Valley (California, USA), 1984.

25. S. Kamin and J.J. Levy, "Attempts for Generalizing the Recursive Path Ordering," Inria, Rocquencourt, to Appear, 1982.

26. C. Kirchner and H. Kirchner, "Implementation of a General Completion Procedure Parameterized by Built-in Theories And Strategies," Rapport Crin 84-R-85, 1984.

27. D. Knuth and P. Bendix, "Simple Word Problems in Universal Algebras," Computational Problems in Abstract Algebra Ed. Leech J., Pergamon Press, pp. 263-297, 1970.

28. D.S. Lankford, "On Proving Term Rewriting Systems Are Noetherian," Report Mtp-3, Math. Dept., Louisiana Tech University, May 1979.

29. P. Lescanne, "Computer Experiments with the REVE Term Rewriting System Generator," in 10th ACM Conf. on Principles of Programming Languages, pp. 99-108, Austin Texas, January 1983.

30. P. Lescanne, "Uniform termination of term rewriting systems - Recursive decomposition ordering with status," Proceedings 9th Colloque les Arbres en Algebre et en Programmation, pp. 182-194, Cambridge University Press, Bordeaux (France), 1984.

31. M. Munoz, "Probleme de terminaison finie des systemes de reecriture equationnels," These 3eme Cycle, Universite de Nancy 1, 1984.

32. M.H.A. Newman, "On Theories With A Combinatorial Definition of <<Equivalence>>," Annals of Math. 43,2, pp. 223-243, 1942.

33. G. Peterson and M. Stickel, "Complete sets of reduction for equational theories with complete unification algorithms," J. of ACM, vol. 28, no. 2, pp. 233-264, 1981.

34. D. Plaisted, "A Recursively Defined Ordering for Proving Termination of Term Rewriting Systems," Dept. of Computer Science Report 78-943, U. of Illinois At Urbana-Champaign, Sept. 1978.

35. D.A. Plaisted, "An associative path ordering," in Proc. NSF Workshop on the Rewrite Rule Laboratory, pp. 123-136, General Electric, Schenectady, New-York, April 1984.

36. J.L. Remy and H. Zhang, "REVEUR 4: a System for Validating Conditional Algebraic Specifications of Abstract Data Types," Proceedings of the 5th ECAI, Pisa, 1984.

37. M. Rusinowitch, "Path of Subterm Ordering and Recursive Decomposition Ordering Revisited," in Proc. 1rst Conf. on Rewriting Techniques and Applications, Lecture Notes in Computer Science, vol. 202, pp. 225-240, Springer Verlag, Dijon (France), 1985.

38. C. Thomas, "RRLab - Rewrite Rule Labor," Memo SEKI 84-01, SEKI-Projekt Fachbereich Informatik Universitat Kaiserslautern, Kaiserslautern RFA, 1984.

NOM DE L'ETUDIANT : GNAEDIG Isabelle

NATURE DE LA THESE : Doctorat 3ème cycle en Informatique

VU, APPROUVE ET PERMIS D'IMPRIMER

NANCY, le 24 JAN. 1988 n°90

LE PRESIDENT DE L'UNIVERSITE DE NANCY I

UNIVERSITE NANCY I

R. MAINARD PRESIDENT

RESUME

L'objet de cette thèse est la réalisation d'un outil de preuve pour la terminaison des systèmes de réécriture équationnels, tout particulièrement dans le cas de la théorie associative-commutative.

Nous fondant sur un théorème général de terminaison équationnelle, nous construisons un ordre sur les termes, possédant les propriétés requises par ce théorème. Puis nous donnons les justifications théoriques des propriétés de notre ordre, apportant ainsi un outil valide de preuve de terminaison des systèmes de réécriture associatifs-commutatifs.

Nous proposons de plus une démonstration complète d'une propriété de l'ordre qui rend cette méthode de preuve implantable dans le cas général des termes avec variables: la stabilité par instanciation,

Nous exposons enfin le champ d'application de l'ordre construit, en expliquant son fonctionnement sur une série d'exemples usuels de systèmes de réécriture associatifs-commutatifs.

MOTS-CLES: réécriture, théorie équationnelle, théorie associative-commutative, terminaison, ordre noethérien