ORSAY
N^o d'ordre: 2821

UNIVERSITÉ DE PARIS-SUD
CENTRE D'ORSAY

THÈSE

présentée
pour obtenir

Le GRADE de DOCTEUR EN SCIENCES
DE L'UNIVERSITÉ PARIS XI ORSAY

PAR

Claude MARCHÉ

SUJET:

Réécriture modulo une théorie présentée par un système convergent et décidabilité du problème du mot dans certaines classes de théories équationnelles

soutenue le premier octobre 1993 devant la commission d'examen

MM.	Gérard HUET	Président
	Leo BACHMAIR	Rapporteurs
	Jacques MORGENSTERN	
	Michaël RUSINOWITCH	
	Bruno BUCHBERGER	Examinateurs
	Dominique GOUYOU-BEAUCHAMPS	
	Jean-Pierre JOUANNAUD	

Réécriture modulo une théorie présentée par un système convergent et décidabilité du problème du mot dans certaines classes de théories équationnelles

Claude Marché

Remerciements

Pour commencer je voudrais remercier mon directeur de thèse, Jean-Pierre Jouannaud. On remercie habituellement son directeur de thèse pour nous avoir donné la possibilité de préparer la thèse, de la confiance qu'il nous a accordée, de son aide et de ses conseils pendant les années de recherche. Cela est bien entendu le cas pour moi, mais je voudrais aussi le remercier pour la liberté et l'autonomie qu'il a su m'accorder dans mon travail, une liberté et une autonomie très larges qui ont particulièrement convenues à mon tempérament.

Je voudrais en second lieu remercier les personnes qui ont accepté d'être membre de mon jury.

Bruno Buchberger a bien voulu participer à ce jury avant même de lire le manuscrit, m'accordant ainsi une grande confiance, et le lecteur s'apercevra facilement en lisant ce mémoire de l'importance de ses travaux antérieurs pour cette thèse, et comprendra pourquoi je suis honoré de cette confiance.

Gérard Huet avait accepté depuis longtemps de présider mon jury, je voudrais l'en remercier et également le saluer pour sa façon très chaleureuse de discuter avec les jeunes chercheurs, que j'ai beaucoup appréciée dans les discussions que j'ai eues avec lui.

Je remercie aussi Jacques Morgenstern qui a accepté d'être rapporteur, après avoir longuement hésité car s'estimant non compétent sur le sujet, ne faisant pas partie de la communauté de la réécriture. Je suis très satisfait qu'il ait finalement accepté, car ses remarques et ses conseils ont été particulièrement utiles à la qualité de la version finale de ce mémoire.

Michaël Rusinowitch avait depuis longtemps accepté d'être rapporteur de ma thèse, et s'était déjà depuis longtemps intéressé à mon travail. Je le remercie de sa confiance, de son aide et de ses conseils dans les discussions malheureusement trop rares que j'ai eues avec lui.

Je remercie également Leo Bachmair pour avoir accepté au dernier moment de faire un rapport sur cette thèse, et je voudrais aussi saluer sa grande rigueur et sa grande compétence scientifique, car plusieurs fois déjà il m'avait signalé des erreurs dans mes démonstrations, et ce malgré leur technicité.

Enfin, je remercie Dominique Gouyou-Beauchamps qui a bien voulu s'intéresser à mon travail bien que n'étant pas spécialiste de la réécriture, et a accepté d'être examinateur de ma thèse avec une très grande gentillesse qui ne surprendra pas ceux qui le connaissent.

En dernier lieu, je voudrais remercier toutes celles et tous ceux qui ont été membres de l'équipe « DEMONS » du L.R.I. pendant les trois années de thèse que j'y ai passé. Alex, Catherine, Delia, Emmanuel, Evelyne, Florent, Hervé, Hubert, Judith, Laurence, Marianne, Maribel, Walid et Zina, je vous remercie pour l'ambiance que vous avez su créer dans cette équipe, et je me souviendrais particulièrement d'une lettre collective extrêmement sympathique reçue à un moment difficile, qui m'est allée droit au cœur.

Table des matières

Introduction

1 Des équations et des sciences

« Ma vie fut partagée entre la politique et les équations. Les équations sont plus importantes pour moi parce que la politique représente le présent, alors qu'une équation est quelque chose d'éternel. » C'est Albert Einstein lui-même qui nous montre dans cette citation la place privilégiée qu'occupent les équations dans les sciences modernes. Il est d'usage de dire que l'on « met en équations » un problème avant de le résoudre, et en effet, la formalisation des problèmes à l'aide d'équations est une méthode très répandue dans toutes les sciences.

L'informatique n'échappe pas à cette règle, et l'on peut énumérer beaucoup de sujets où les équations joue un rôle essentiel : spécifications algébriques, prototypage, programmation logique avec égalité, programmation fonctionnelle, λ-calcul (logique des combinateurs), calcul formel, optimisation de code, inférence de type, géométrie algorithmique, etc.

À quoi servent les équations ? Une fois qu'un problème est mis en équations, il y a plusieurs questions auxquelles on peut s'intéresser : un premier type de question auquel on s'intéresse est de *résoudre* des équations, c'est-à-dire trouver une solution (ou toutes les solutions), une solution étant la donnée pour chaque variable d'une valeur rendant vraie l'équation. Par exemple, on sait bien résoudre une équation du second degré $ax^2 + bx + c = 0$. Ce problème peut être nommé d'une façon générale, dans le cadre symbolique où nous nous placerons, le problème de *satisfiabilité* (avec comme cas particulier *l'unification*).

Un second type de problème est de savoir si une équation est *valide*, autrement dit est-ce que l'équation est satisfaite pour *toutes* les valeurs possibles des variables. Il est usuel d'appeler de telles équations des *identités*. Par exemple, chacun connaît l'identité remarquable $(x+y)^2 = x^2 + 2xy + y^2$ dans les anneaux commutatifs. Ce second type de problème s'appelle problème de la *validité* ou de *l'identité*, ou encore *problème du mot* dans le cas d'identités sans variables, les deux problèmes étant en fait équivalents dans le cas qui nous intéresse (à savoir, l'interprétation dans la classe de tous les modèles équationnels).

Le *problème du mot* d'une théorie équationnelle (définie par un ensemble d'équations E) est le problème de savoir si deux expressions (deux « termes » dans notre terminologie) sont égales à partir du moment où les équations de E sont admises. Par exemple, on peut chercher à montrer que dans tout groupe, c'est-à-dire dans toute structure satisfaisant les équations des groupes (associativité, élément neutre et inverse), on a l'identité $(xy)^{-1} = y^{-1}x^{-1}$. Un autre exemple particulièrement intéressant pour le génie logiciel est la preuve de propriétés de spécifications algébriques, écrites sous forme d'équations.

En fait, la question qui nous intéresse plus particulièrement est la décision *automatique* du problème du mot. Y a-t-il un moyen *mécanique* de savoir si l'égalité de deux termes est conséquence d'un ensemble d'axiomes ?

En 1900, David Hilbert, parmi ses célèbres problèmes, avait posé la question de savoir s'il existait un tel moyen *mécanique* pour engendrer tous les théorèmes mathématiques. La première réponse à cette question a été donnée par Kurt Gödel en 1931, avec son non moins célèbre théorème d'incomplétude, qui montrait l'existence dans toute logique cohérente au moins aussi puissante que l'arithmétique de Peano de propositions *indémontrables*, c'est-à-dire qui ne soit pas conséquence des axiomes, et sa négation non plus. Ensuite, en 1936, Alan Turing, grâce à ses célèbres machines, a montré qu'il existait des problèmes *indécidables*, c'est-à-dire auquel aucun moyen mécanique ne peut donner de réponse (en fait, sous réserve d'admettre la thèse de Church). Ceci répond donc négativement à la question de Hilbert, mais ne résout pas encore le problème si l'on se restreint uniquement à la logique équationnelle, où le seul prédicat est l'égalité.

En 1947, Emil L. Post et A. A. Markov ont indépendamment prouvé que le problème du mot dans les théories équationnelles est indécidable, répondant donc également négativement à la question de Hilbert dans le cas des théories équationnelles. Pour plus de précision sur les problèmes de mots, on pourra se référer à l'ouvrage récent de E. Börger [15]. Ce n'est pas pour autant qu'il n'y a plus rien à faire : ce que nous allons essayer de faire, c'est de décider le problème du mot dans certaines classes de théories équationnelles.

2 De la réécriture et du calcul de formes normales

L'outil que l'on va utiliser pour tenter de décider le problème du mot est la *réécriture*. Il est difficile de dire qui est à l'origine de la notion de réécriture, mais on peut citer Post et Thue pour ce qui concerne la réécriture sur les mots. On trouvera une discussion complète et beaucoup de références dans l'article de synthèse de Nachum Dershowitz et Jean-Pierre Jouannaud [30]. Il semble que les pionniers dans l'utilisation de la réécriture comme technique de décision de l'égalité soient M. Newman [74], T. Evans [36] et Donald Knuth [55].

La réécriture doit être comprise comme une stratégie *ordonnée* d'utilisation de l'égalité : on utilise les équations dans un sens et pas dans l'autre. Afin que cette technique soit intéressante, il faut exiger deux propriétés sur les règles qui sont la *terminaison* et la *confluence*. Pour introduire la notion de réécriture, considérons le problème suivant, extrait d'une olympiade de mathématiques.

Sur une île vivent des caméléons qui ont le comportement bien connu des caméléons à savoir qu'ils ont la capacité d'adapter leur propre couleur à leur environnement. Le métabolisme de ces créatures leur permet d'opter à volonté pour l'une des couleurs rouge, vert ou bleu. Lorsque deux caméléons de couleur différente se rencontrent, ils essaient simultanément de prendre la couleur de l'autre, et ils se retrouvent tous deux de la troisième couleur ! Un biologiste étudiant ce phénomène est amené à se poser la question suivante : « Depuis le temps que cette colonie de caméléons évolue sans interaction avec l'extérieur, comment se fait-il qu'ils ne soient pas devenus tous de la même couleur ? » Afin de préciser ses analyses, notre biologiste dénombre les caméléons à un instant donné et trouve exactement 15 caméléons rouges, 14 bleus et 13 verts. Nous verrons plus tard comment ces données suffisent pour répondre à la question posée, et le lecteur est invité à y réfléchir.

Ce à quoi nous allons nous intéresser pour l'instant est de formaliser l'évolution de la colonie de caméléons en terme de réécriture. Les trois changements de couleur possibles sont définis par les « règles » suivantes :

$$R \left\{ \begin{array}{l} \text{rouge} + \text{vert} \rightarrow \text{bleu} + \text{bleu} \\ \text{rouge} + \text{bleu} \rightarrow \text{vert} + \text{vert} \\ \text{bleu} + \text{vert} \rightarrow \text{rouge} + \text{rouge} \end{array} \right.$$

L'évolution de la colonie à partir d'une situation donnée est modélisé par l'application de ces règles, et comme les rencontres se produisent de manière aléatoire, l'application des règles est *non déterministe*.

Prenons maintenant quelques exemples. Supposons d'abord que la situation à un moment donné est vert + vert + bleu, l'évolution sera alors (en fait, dans ce cas, il n'y a qu'une possibilité à chaque fois) :

$$\begin{array}{rcl} \text{vert} + \text{vert} + \text{bleu} & \underset{R}{\longrightarrow} & \text{vert} + \text{rouge} + \text{rouge} \\ & \underset{R}{\longrightarrow} & \text{bleu} + \text{bleu} + \text{rouge} \\ & \underset{R}{\longrightarrow} & \text{bleu} + \text{vert} + \text{vert} \\ & \underset{R}{\longrightarrow} & \text{vert} + \text{rouge} + \text{rouge} \\ & \text{etc.} & \end{array}$$

On voit que, dans ce cas particulier, on n'atteint jamais une situation stable, c'est-à-dire unicolore, dans laquelle il n'y a plus d'évolution possible. Lorsque, comme cet exemple, un système de règles autorise des séquences infinies de transformations, on dit que ce système *ne termine pas*. Imaginons que l'on supprime la troisième règle du système R, alors le système de réécriture obtenu cette fois termine : à chaque fois qu'une règle est appliquée, un rouge est effacé, ceci ne peut se produire qu'un nombre fini de fois. Cette propriété de terminaison va être fondamentale pour nous, car elle assure que les calculs ne vont pas durer indéfiniment. Lorsque un terme n'est réductible par aucune règle, c'est-à-dire pour les caméléons toutes les configurations unicolores, il est dit en *forme normale*. Lorsqu'un système de règles termine, on est assuré que tout terme possède (c'est-à-dire se réécrit en) au moins une forme normale.

Regardons maintenant un autre exemple de configuration initiale pour les caméléons, la situation rouge + vert + bleu. Sur cette configuration, chacune des règles peut en fait être appliquée, on a par exemple

$$\text{rouge} + \text{vert} + \text{bleu} \underset{R}{\longrightarrow} \text{bleu} + \text{bleu} + \text{bleu}$$

et

$$\text{rouge} + \text{vert} + \text{bleu} \underset{R}{\longrightarrow} \text{rouge} + \text{rouge} + \text{rouge},$$

et les deux termes obtenus sont cette fois irréductibles. On constate sur cet exemple-ci que l'on peut obtenir, à partir d'un même terme, deux formes normales distinctes.

Afin d'assurer qu'il n'existe pas plusieurs formes normales distinctes, nous allons nous intéresser aux systèmes qui possèdent la propriété de *confluence*, qui exprime que si deux termes peuvent être obtenus à partir d'un même terme par des suites de réductions différentes, on peut toujours les réduire encore en un même terme :

$$\begin{tikzcd}[column sep=small] & t \arrow[dl, "*"', "R"] \arrow[dr, "*", "R"'] & \\ s & & u \end{tikzcd} \quad \Longrightarrow \quad \exists v \quad \begin{tikzcd}[column sep=small] s \arrow[dr, "*", "R"'] & & u \arrow[dl, "*"', "R"] \\ & v & \end{tikzcd}$$

Pour résoudre le problème du mot pour une théorie équationnelle donnée, nous allons chercher à construire un système de règles équivalent convergent (c'est-à-dire qui termine et qui soit confluent). Ces deux propriétés impliquent l'existence d'une forme normale unique pour chaque terme, et alors l'égalité de deux termes dans la théorie est équivalente à l'égalité syntaxique de leurs formes normales.

Nous verrons que nous pourrons construire un système convergent pour la théorie des caméléons. Comme exemple de système convergent, qui montre bien pourquoi la propriété de forme normale unique rend le résultat de la réécriture déterministe, on peut donner la dérivation des fonctions usuelles en mathématiques :

$$\begin{cases} \partial_x(f+g) \to \partial_x(f) + \partial_x(g) \\ \partial_x(f.g) \to \partial_x(f).g + f.\partial_x(g) \\ \partial_x(f \circ g) \to (\partial_x(f) \circ g).\partial_x(g) \\ \partial_x(\sin x) = \cos x \\ \partial_x(\ln x) = \frac{1}{x} \\ \partial_x(a) = 0 \\ \partial_x(x) = 1 \\ etc. \end{cases}$$

Les logiciels de calcul formel utilisent effectivement des systèmes de réécriture pour calculer les dérivés des expressions construites à partir des fonctions usuelles, mais également pour d'autres opérations algébriques : simplification, factorisation ou développement, sur des fractions rationnelles, des séries formelles, des expressions contenant des fonctions usuelles (exponentielles et logarithmes, fonctions trigonométriques), formules trigonométriques.

3 De la déduction automatique et de la complétion

L'idée essentielle qui va nous guider est la suivante : à partir d'un ensemble d'équations, essayer de construire un ensemble de règles de réécriture tel que deux termes soient équivalents si et seulement s'ils se réécrivent en un même terme. Ce mécanisme, appelé la *complétion* a été introduit en 1970, par Donald Knuth et Peter Bendix [55]. La complétion telle qu'ils l'ont définie consiste essentiellement en deux transformations.

La première de ces opérations est l'orientation des équations en règles. Le sens dans lequel on oriente doit être judicieux pour assurer la terminaison du système ainsi obtenu : la méthode standard pour réaliser cela est de définir une relation d'ordre sur les termes qui permette de comparer les deux membres d'une équation. Pour garantir la terminaison, un tel ordre devra être noethérien, c'est-à-dire ne pas admettre l'existence de suites infinies strictement décroissantes de termes. Cette idée qui semble naturelle n'est malheureusement pas suffisante pour garantir notre but, qui est rappelons-le de produire un ensemble de règles tel que deux termes doivent être équivalents si et seulement s'ils se réécrivent en un même terme.

Il suffit d'un exemple très simple pour voir que l'orientation seule n'est pas une transformation suffisante : si l'on part de deux équations $a = b$ et $a = c$, que l'on oriente en $a \to b$ et $a \to c$, les deux termes b et c sont équivalents mais ils ne peuvent pas se réécrire en un même terme. Dans cet exemple, on dira que $\{b, c\}$ est une *paire critique*. La deuxième opération mise en jeu dans la complétion est ainsi le *calcul des paires critiques*, et l'ajout de ces paires comme nouvelles équations. Cette phase s'appelle encore *déduction* puisqu'elle déduit des équations nouvelles à partir des anciennes. Le résultat fondamental qui fait que ce mécanisme fonctionne est que le nombre de déductions nécessaires est *fini*. Le résultat appelé *lemme des paires critiques*, dit que la confluence est assurée à partir du moment où les paires critiques (qui sont en nombre fini) sont *joignables*, c'est-à-dire les deux termes de la paire se réécrivent en un même terme par le système.

Bien que le processus de déduction n'engendre qu'un nombre fini de paires critiques à partir de deux règles données, la complétion peut *diverger*, c'est-à-dire calculer indéfiniment de nouvelles paires critiques. En effet, les paires critiques ainsi engendrées sont ajoutées comme nouvelles équations, qui seront orientées et peuvent donner lieu à de nouvelles paires critiques ! Autrement dit, les transformations d'orientation et de déduction terminent séparément, mais leur mélange ne termine pas.

Le lemme des paires critiques assure que lorsque la complétion termine, le système de règles obtenu est convergent, car l'algorithme a calculé toutes les paires critiques entre les règles du système, et les a jointes. Un résultat plus général de *complétude* a été prouvé en 1981 par Gérard Huet [42] qui s'applique aussi dans le cas où la complétion diverge, à condition d'avoir une hypothèse supplémentaire dite *d'équité*. Ce dernier résultat montre alors que la complétion est un algorithme de semi-décision de la théorie : en donnant l'équation à démontrer à réécrire « en parallèle » avec la complétion, c'est-à-dire si l'on normalise les deux membres de l'équation chaque fois qu'une nouvelle règle est produite, on est sûr que si l'équation est un théorème, elle sera prouvée en temps fini, par contre si elle est fausse on n'obtiendra jamais de réponse.

En 1986, Leo Bachmair, Nachum Dershowitz et Jieh Hsiang [8] ont introduit la technique d'algébrisation des preuves, et ont décrit l'algorithme de Knuth et Bendix par, d'une part un système de *règles d'inférence*, et d'autre part une *stratégie* d'application de ces règles. Cette présentation donne lieu à un formalisme beaucoup plus agréable pour la preuve de complétude : la méthode dite *de réduction des preuves*. Cette méthode sera celle que l'on va utiliser dans ce mémoire pour montrer nos résultats de complétude.

4 Des équations inorientables et des extensions

L'exemple fondamental de complétion donné par Knuth et Bendix est la construction d'un système de réécriture convergent pour la théorie des groupes. Cette théorie est présentée par les équations suivantes (en fait, on peut se contenter de trois équations, mais c'est plutôt sous cette forme que l'on apprend ce que sont les groupes) :

$$\begin{cases} (x.y).z = x.(y.z) \\ x.1 = 1.x = x \\ x.x^{-1} = x^{-1}.x = 1 \end{cases}$$

La complétion de Knuth et Bendix produit l'ensemble des dix règles suivantes :

$$\left\{\begin{array}{l} (x.y).z \rightarrow x.(y.z) \\ x.1 \rightarrow x \\ 1.x \rightarrow x \\ x.x^{-1} \rightarrow 1 \\ x^{-1}.x \rightarrow 1 \\ (x.y)^{-1} \rightarrow y^{-1}.x^{-1} \\ (x^{-1})^{-1} \rightarrow x \\ 1^{-1} \rightarrow 1 \\ x.(x^{-1}.y) \rightarrow y \\ y^{-1}.(y.x) \rightarrow x \end{array}\right.$$

Malheureusement, si l'on essaie de compléter la théorie des groupes abéliens, c'est-à-dire en ajoutant l'équation de commutativité $x.y = y.x$, alors l'algorithme échoue. La raison est que l'équation de commutativité est essentiellement inorientable : quel que soit le sens choisi, la règle obtenue peut engendrer des suites infinies de réductions. Cet exemple très simple montre la limitation de la complétion de Knuth-Bendix.

Pour pouvoir traiter l'équation de commutativité, Dallas Lankford et A. Ballantyne [61] (pour les monoïdes commutatifs finiment engendrés), et indépendamment Gerald Peterson et Mark Stickel [78] ont proposé d'étendre la notion de réécriture à la réécriture dite *équationnelle*.

L'idée est d'intégrer une équation inorientable, comme la commutativité, dans le processus de réécriture, le filtrage étant remplacé par le filtrage *modulo* cette théorie équationnelle. Peterson et Stickel ont alors généralisé l'algorithme de Knuth et Bendix à la réécriture équationnelle modulo AC, c'est-à-dire Associativité et Commutativité (il faut remarquer que si l'on intègre la commutativité dans le processus de réécriture, alors l'associativité ne s'oriente plus non plus ! il faut donc l'intégrer aussi...). Ils ont montré qu'il fallait pour ce faire remplacer le calcul des paires critiques par les paires critiques *modulo AC*, qui sont calculées par *unification AC* plutôt que par l'unification traditionnelle. De plus, il faut rajouter une règle d'inférence supplémentaire pour avoir la complétude, la règle *d'extension*. Étendre une règle de la forme $s + t \rightarrow u$ consiste simplement à rajouter la règle $x + s + t \rightarrow x + u$, x étant une variable n'apparaissant pas déjà dans s, t ou u. L'algorithme de complétion ainsi défini, dit *de complétion associative-commutative*, est alors suffisamment puissant pour calculer un système convergent pour les groupes abéliens.

L'algorithme de complétion AC de Peterson et Stickel est suffisamment puissant pour calculer des systèmes convergents pour les structures mathématiques classiques, comme on l'a déjà vu les groupes abéliens, mais aussi les anneaux commutatifs, les anneaux booléens, ou encore la théorie des homomorphismes sur ces structures, les autres équations que l'on introduit étant toutes orientables d'une façon raisonnable. Mais une chose que l'on peut reprocher à cet algorithme de complétion est sa grande complexité, nettement plus élevée que la complétion de Knuth et Bendix à cause de l'unification AC, elle peut engendrer une quantité beaucoup plus grande de paires critiques à chaque étape. En effet, alors que l'unification standard n'engendre pour chaque équation qu'un seul unificateur principal, et donc qu'une seule paire critique, l'unification AC peut engendrer une grande quantité de paires critiques à partir d'une seule superposition donnée entre deux règles, un nombre qui est *a priori* une fonction doublement exponentielle de la taille des termes à unifier.

5 Des contraintes et des instances interdites

En 1988, Timothy Baird, Gerald Peterson et Ralph Wilkerson ont eu l'idée de remplacer la réécriture modulo AC par la réécriture modulo AC1, où le « 1 » signifie l'équation d'existence d'un élément neutre ($x + 0 = x$). Le but ne consistait pas, bien sûr, à résoudre un problème d'orientation (l'équation d'élément neutre s'orientant très naturellement en $x + 0 \to x$), mais à remplacer dans l'algorithme de complétion l'unification AC par l'unification AC1, qui était connue comme ayant de meilleurs propriétés que l'unification AC : le nombre de solutions d'une équation donnée est « seulement » simplement exponentiel en fonction de la taille des termes à unifier, en fait parce qu'il y a toujours une solution unique en l'absence de constantes libres, alors que le nombre de solutions modulo AC est doublement exponentiel.

À cette époque avaient déjà été présentés des algorithmes de complétion modulo une théorie E arbitraire, mais vérifiant quand même quelques propriétés. Il y eût d'abord Jean-Pierre Jouannaud et Hélène Kirchner, en 1984 [43], qui avaient besoin de supposer les classes modulo E *finies* : cette condition n'est pas vérifiée pour AC1 ! Enfin, en 1987 [7], répondant positivement à la conjecture de Jouannaud et Kirchner, Bachmair et Dershowitz ont supprimé cette hypothèse et ont donné un algorithme de complétion modulo E, sous l'hypothèse moins forte que l'ordre de sous-terme modulo E soit noethérien, hypothèse qui n'est pas satisfaite non plus dans le cas de AC1 !

Le problème vient de ce que la réécriture modulo AC1 peut ne pas terminer à cause de l'utilisation du filtrage AC1, pour des règles pourtant élémentaires. Dans le cas de notre exemple des groupes abéliens, la règle qui calcule l'opposé d'une somme $-(x + y) \to (-x) + (-y)$ ne termine pas modulo AC1, puisque l'on a la suite infinie de réductions suivantes :

$$\begin{array}{lll} -0 & =_{AC1} -(0+0) & \to (-0)+(-0) \\ & =_{AC1} -(0+0)+(-0) & \to (-0)+(-0)+(-0) \\ & etc. & \end{array}$$

L'idée de Baird, Peterson et Wilkerson a été d'utiliser la réécriture *contrainte*, où à une règle est associée une contrainte, qui n'autorise la règle à s'appliquer que lorsque cette contrainte est satisfaite. Par exemple, pour l'inverse d'une somme, on utilise la règle contrainte $-(x + y) \to (-x) + (-y) \mid x \neq 0 \wedge y \neq 0$. Baird, Peterson et Wilkerson ont alors présenté un algorithme de complétion modulo AC1 [11], mais malheureusement leur algorithme n'était pas complet, il manquait certaines déductions éventuelles induites par l'existence d'un élément neutre.

Jean-Pierre Jouannaud et moi-même avons rétabli la complétude grâce au calcul *d'instances interdites* associées à chaque règle [45]. Le problème avec cette technique de réécriture contrainte est qu'elle ne se généralise pas facilement à d'autres théories. Par exemple, on aurait voulu utiliser la même technique pour ACI (où « I » désigne l'équation d'idempotence $x + x = x$), mais cela n'est pas possible car en fait, pour toute règle $l \to r$, la réécriture modulo ACI par $l \to r$ ne termine pas, ceci quelle que soit l'instance utilisée :

$$\begin{array}{lll} l & =_{ACI} l+l & \to l+r \\ & =_{ACI} l+l+r & \to l+r+r \\ & etc. & \end{array}$$

La méthode de réécriture contrainte semble donc conduire à un cul-de-sac, et le problème de la complétion modulo ACI est un problème resté ouvert à ce jour, Nous montrons dans cette thèse comment le résoudre.

Il faut bien remarquer que la réécriture contrainte a bien sûr un intérêt mais pour d'autres raisons, par exemple dans les algorithmes de complétion par stratégie basique [9, 75], mais aussi pour généraliser les techniques équationnelles à des domaines justement non équationnels, contraintes numériques par exemple. En fait, dans le cas de la complétion AC1, la réécriture contrainte était utilisée pour résoudre essentiellement un problème de terminaison, sans changer la sémantique même de la spécification équationnelle (pour une règle $l \to r \mid c$, toutes les instances de l'équation $l = r$ sont valides, même celles qui ne satisfont pas la contrainte c). Par contre, il est intéressant d'utiliser les contraintes pour étendre le pouvoir d'expression : contraintes d'appartenance pour le typage, contraintes équationnelles pour la stratégie basique, contraintes diséquationnelles pour la réécriture AC1, contraintes d'ordres pour les stratégies ordonnées. La réécriture contrainte ne doit pas être confondue avec la réécriture conditionnelle : par exemple, pour définir la théorie des corps, avec la règle conditionnelle $x.x^{-1} \to 1 \mid x \neq 0$, où $\neq$ est interprété par la négation de l'égalité modulo la théorie définie par les équations conditionnelles elle-mêmes.

6 Des polynômes et des bases standard

Un résultat antérieur à la complétion de Knuth-Bendix, dû à Bruno Buchberger [19] en 1965, est l'existence et le calcul de bases standard, dites encore bases de Gröbner, d'idéaux de polynômes : étant donné un anneau commutatif A et un idéal I de l'anneau de polynômes à une ou plusieurs indéterminées $A[X_1, \dots, X_n]$, une *base standard* de I est une famille de polynômes $P_1, \dots, P_k$ vérifiant des propriétés analogues aux bases des espaces vectoriels. Buchberger a décrit un algorithme qui à partir de n'importe quelle famille génératrice de I calcule une base standard de I, lorsque A est un corps. Ensuite, des extensions ont été données par Abdelilah Kandri-Rody et Deepak Kapur dans le cas où A est un anneau euclidien [48, 51].

Ce qui nous intéresse dans ces résultats est qu'il est facile de faire correspondre à un polynôme une équation, ou une règle, de telle façon que le calcul d'une base standard d'un idéal engendré par des polynômes $(P_1, \dots, P_k)$ sur $\mathbb{Z}[X_1, \dots, X_n]$ correspond au calcul d'un système convergent (par une technique proche de la complétion, mais il faut bien remarquer que cet algorithme est *antérieur* à la complétion de Knuth-Bendix) pour la théorie équationnelle définie par les équations de la théorie des anneaux commutatifs et les équations associées aux P_i, qui peuvent être n'importe quelles équations closes (c'est-à-dire sans variables). Autrement dit, le calcul des bases standard est une technique possible de décision du problème du mot dans les théories définies par les équations des anneaux commutatifs et des équations closes arbitraires (du moins lorsque la signature ne contient que des constantes).

Les algorithmes de Buchberger et de Kandri-Rody et Kapur sont proches de la complétion car ils consistent à fabriquer des règles à partir des polynômes, ce qui correspond à l'orientation, et à calculer des nouveaux polynômes appartenant à l'idéal, ce qui correspond aux paires critique. De nombreux chercheurs se sont intéressé à essayer « d'unifier » ces deux algorithmes, c'est-à-dire décrire un seul et même algorithme dont à la fois la complétion de Knuth-Bendix (et ses extensions) et les algorithmes de calcul de bases standard soient des instances particulières, mais à notre point de vue personne n'a jusqu'à présent trouvé de solution vraiment satisfaisante. Nous nous bornerons ici à citer quatre travaux, bien qu'il y en ai eu certainement de multiples autres, et le lecteur pourra se reporter à un article de synthèse de Buchberger [20].

Tout d'abord, dans l'ordre chronologique on trouve les travaux de Philippe Le

Chenadec [62] : il s'est intéressé à décrire des algorithmes de complétion dans des algèbres particulières, par exemple les groupes finiment engendrés. Dans chacune des algèbres auxquelles il s'est intéressé, il définit une relation de réduction appropriée et donne une procédure de complétion qui en résulte. La notion d'orientation est adaptée à chacun des cas, avec ce que Le Chenadec appelle la règle de symétrisation. Cette technique ne s'applique qu'aux systèmes clos. La technique de symétrisation qu'il introduit est très intéressante et nous l'utilisons dans ce mémoire en la généralisant au cas non clos.

Ensuite, nous citerons les travaux de Kandri-Rody, Kapur, Narendran et Winkler [49], qui ont défini une réduction modulo une « relation de simplification » T, mais la critique que l'on peut faire de ce travail est que toutes les déductions entre R et T ne sont pas calculées. Il se trouve que ca marche dans les cas particuliers qu'ils considèrent, mais c'est la même erreur que Baird, Peterson et Wilkerson, ce n'est pas complet dans le cas général. C'est quand même cette idée que nous avons repris pour définir la réécriture normalisée qui est effectivement une réécriture modulo une « relation de simplification » : la relation de réduction définie par un système convergent. Nous corrigeons l'incomplétude en introduisant la notion d'instances critiques et de couples normalisants.

Une approche très différente de ce problème de la relation entre la complétion de Knuth-Bendix et le calcul de bases standard a été utilisée par Loïc Pottier [81] : il s'est placé d'un point de vue très abstrait en montrant que si l'on a un ensemble muni d'opérations abstraites de réécriture et de paires critiques, vérifiant de bonnes hypothèses, alors on a un algorithme de complétion. La critique que l'on peut faire de ce travail est qu'il n'y pas d'algorithme général de complétion, dont la complétion de Knuth-Bendix et le calcul de bases standard soient tous deux des instances. Notre approche est plus satisfaisante car nous montrons que la relation entre les deux algorithmes est plus étroite : aussi bien les termes que les polynômes sont des instances du même type d'objets, en l'occurrence des éléments d'une algèbre quotient. La relation de réduction est dans les deux cas la réécriture S-normalisée pour un choix particulier de S, et enfin l'orientation et le calcul des paires critiques sont dans les deux cas une instance de l'orientation et du calcul de paires critiques via le calcul des couples normalisants. Nous donnons ainsi effectivement un algorithme de complétion générique paramétré par un système de réécriture convergent S, avec une notion de paires critiques (avec les couples normalisants) et qu'ensuite seulement nous remarquons que l'on obtient comme cas particuliers la complétion de Knuth-Bendix et les algorithmes de calcul de bases standard lorsque S est bien choisi. Autrement dit, nous donnons un algorithme général satisfaisant les hypothèses retenues par Pottier. Par contre, un avantage de l'approche de Pottier est qu'elle s'applique également aux cas des corps.

Enfin, nous citerons le travail de Reinhard Bündgen [21], qui a montré que l'on peut « simuler » l'algorithme de Buchberger avec un algorithme de complétion AC auquel on donne en entrée les équations définissant la théorie des anneaux commutatifs, et on ajoute les équations closes définies par les polynômes engendrant l'idéal, les indéterminées étant des constantes. Si l'on lance la complétion sans rien dire de plus, la quantité de paires critiques à calculer croît extrêmement vite avec le nombre de polynômes donnés, et en pratique on ne peut pas obtenir le résultat attendu (voir les exemples chapitre 3). Les « simulations » consistent en fait à ajouter des heuristiques : priorités d'application de règles, élimination de paires critiques, etc. De plus, la technique de Bündgen ne fonctionne réellement que dans le cas des corps finis (qui contrairement à la théorie des corps en général, admettent une présentation équationnelle finie), il ne simule pas l'algorithme de calcul de bases standard pour les anneaux euclidiens.

En fait, la différence fondamentale entre toutes ces approches et la nôtre (qui est également celle de Narendran et Rusinowitch dans le cas AC-clos) est que notre approche se place sur des algèbres universelles en général, alors que l'approche précédente consistait à s'intéresser à des algèbres particulières (groupes, anneaux, modules, avec commutativité ou non, etc.), et à traiter chacun des cas séparément par des techniques qui bien sûr ont des analogies entre elles, mais sont adaptées à chaque cas particulier. Notre approche consiste à présenter un cadre général pour lequel ont trouve comme instance les cas particuliers. Nous n'avons pas cherché *a priori* à « simuler » les algorithmes de calcul de bases standard par de la complétion, nous avons décrit un algorithme très général, la complétion S-normalisée, qui permet aussi bien de calculer des systèmes de réécriture canoniques comme le fait la complétion de Knuth-Bendix que de calculer des bases standard, suivant le choix de S.

Nous reprenons en fait l'idée de Kandri-Rody, Kapur, Narendran et Winkler de la réécriture modulo une relation de simplification, où cette relation est la normalisation par un système canonique, mais nous ne supposons rien sur ce système, et nous devons bien faire attention de calculer les déductions entre les équations que l'on a en entrée et les règles de ce système canonique, sous la forme de *couples normalisants*, notion qui généralise la notion d'instances critiques que nous avions introduite pour la complétion AC1-contrainte.

7 Des théories E-closes décidables et d'autres indécidables

Indépendamment (souvent antérieurement) des progrès réalisés dans les algorithmes de complétion, un certain nombre de résultats ont été démontrés concernant les problèmes du mot pour des classes de théories équationnelles particulières.

Le premier de ces résultats est l'indécidabilité du problème du mot sur les *mots* sur un certain alphabet, qui date de 1947 et à été obtenu indépendamment par Post et Markov [67,80]. Les mots peuvent être considérés comme des termes sur une signature formée par ajout d'un symbole binaire à l'alphabet, l'opérateur de concaténation. Cet opérateur vérifie l'axiome d'associativité, ce qui signifie qu'un ensemble d'équations sur les mots définit une théorie close modulo l'associativité, et le résultat de Post et Markov implique ainsi que le problème du mot A-clos est indécidable. Pour être précis, non seulement le problème du mot *uniforme* est indécidable, mais également il existe une théorie A-close particulière dont le problème du mot est indécidable, et J. Matijasevic en a donné un exemple explicite [68].

D'autre part, on sait depuis 1954 par Ackermann [1] que le problème du mot pour les théories closes est décidable, et par le calcul des bases standard de polynômes à coefficients dans $\mathbb{Z}$ que le problème du mot E-clos est décidable si E est présentée par les équations des anneaux (du moins dans le cas où il n'y a que des constantes). La conséquence de ces résultats qui nous intéresse plus particulièrement, c'est que le statut (décidable ou indécidable) du problème du mot dans les théories closes modulo une théorie E donnée peut varier de façon complexe en fonction de E : décidable si E est vide, devient indécidable si E est l'associativité d'un symbole binaire, redevient décidable si E consiste en la théorie des anneaux commutatifs dans son ensemble. Un des points qui nous ont intéressés dans ce mémoire est de regarder ce qui se passe dans les cas de sous-théories des anneaux intermédiaires.

Partons de E contenant l'associativité de $+$, et ajoutons successivement les axiomes définissant les anneaux commutatifs. Commençons par rajouter la commutativité de $+$: on obtient alors la classe des théories AC-closes pour un symbole AC, et si l'on suppose que la signature ne contient, à part $+$, que des constantes, il s'agit de la classe des semi-groupes commutatifs finiment engendrés. La décidabilité du problème du mot dans ces algèbres est connue depuis 1963 par V. Emiličev [35,65,69], bien qu'en fait cela peut être considéré comme un cas particulier de résultats nettement plus anciens [40].

On s'est intéressé depuis à regarder si le problème reste décidable si, d'une part on accepte des symboles non constants dans la signature, et d'autre part si l'on autorise plusieurs symboles AC : la réponse est oui, et a été prouvé en 1991, partiellement par moi-même [66], et (parallèlement) complètement par Paliath Narendran et Michaël Rusinowitch [71]. Ce dernier résultat a nécessité la définition d'un *ordre de réduction AC total* et ensuite la décidabilité est obtenue par un algorithme de complétion « spécialisé » au cas AC-clos, dont on montre la terminaison pour toute donnée. Bien que depuis Eric Domenjoud et Francis Klay aient trouvé une méthode se passant d'un tel ordre pour obtenir le même résultat, nous en aurons nous, encore besoin pour nos complétions E-closes. Nous consacrons à ces ordres le chapitre 2, où nous donnons également quelques nouveautés, dont nous avons besoin pour traiter le cas de la complétion close modulo les anneaux commutatifs.

Si à partir de E =A, au lieu d'ajouter la commutativité, on ajoute plutôt les équations d'élément neutre et d'inverse, afin de définir la théorie des groupes, alors on obtient la classe des théories closes modulo les groupes, qui contient tous les groupes finiment engendrés. Il est connu depuis 1955 que le problème du mot (même celui non-uniforme) est indécidable dans cette classe, cela a été prouvé indépendamment par P. Novikov et W. Boone [14, 77, 85].

Ce phénomène confirme donc l'instabilité du statut du problème du mot E-clos en fonction de E. Dans ce mémoire, nous nous intéresserons à d'autres cas intermédiaires entre E vide et E définissant les anneaux commutatifs; nous regardons aussi le cas des anneaux booléens. Le chapitre 5 seront consacré aux résultats de décidabilité, qui seront obtenus par construction d'un système normalisé convergent, et le chapitre 6 sera consacrés aux résultats d'indécidabilité.

8 Des règles normalisées et des équations critiques

En regardant l'utilisation des instances interdites dans AC1, et de la méthode « d'orientation » dans l'algorithme de Buchberger, ainsi que la méthode qu'il faudrait utiliser pour assurer la terminaison dans ACI, nous avons été amené à introduire la réécriture normalisée. Cette notion est à la confluence des problèmes de terminaison de AC1 et ACI, et de la méthode de l'algorithme de Buchberger. L'idée essentielle est que l'on travaille sur une algèbre quotient et que l'on ne désire travailler que sur les termes en forme normale pour cette algèbre. Les formes normales seront définies par un système convergent éventuellement modulo AC, et donc on ne regardera que les théories E pour lesquels il existe un tel système, ce qui est le cas dans tous les exemples qui précèdent (groupes et anneaux en particulier).

En fait l'idée intuitive est la suivante : l'orientation dans la complétion AC1-contrainte fabriquait à partir d'une équation une règle et un ensemble d'équations; et dans l'algorithme de calcul de bases standard l'orientation produit à partir d'une équation un ensemble de règles. La règle d'orientation que nous introduisons consiste alors de façon générale à transformer une équation en un couple formé d'une part d'un ensemble de règles, dites règles normalisées, et d'autre part d'un ensemble d'équations, dites équations critiques. Nous donnons d'abord au chapitre 4 un système de règles d'inférence pour la complétion normalisée, dont nous montrons la complétude en toute généralité. Ensuite, nous montrons que l'on a comme instance à la fois la notion d'orientation de la complétion standard, la notion d'orientation avec calcul d'instances critiques de la complétion AC1, la notion d'orientation avec symétrisation de Le Chenadec, et la notion d'orientation par calcul du monôme dominant dans l'algorithme de Buchberger, répondant ainsi, pour la première fois à notre avis, de façon satisfaisante au problème de la réunification des algorithmes de complétion de Knuth et Bendix et de calcul de bases standard de Buchberger.

La réécriture et la complétion normalisées seront traitées au chapitre 4.

Enfin, nous montrons des résultats de terminaison de la complétion pour plusieurs classes de théories équationnelles, incluant le cas clos modulo les anneaux commutatifs, donc le calcul des bases standard; et le cas AC-clos; ceci par une méthode en grande partie uniforme. Nous montrons ainsi la grande similitude qui existe entre le résultat de terminaison de Buchberger et la terminaison de la complétion dans le cas clos. Ceci sera traité au chapitre 5.

9 Implantation

Afin de tester expérimentalement la technique de complétion normalisée, nous avons implanté cet algorithme, avec la possibilité de choisir la théorie S modulo laquelle on travaille parmi un ensemble de théories prédéfinies : élément neutre, idempotence, nilpotence, groupes abéliens, anneaux commutatifs et corps finis. Cette implantation est décrite en annexe.

Chapitre 1

Définitions de base

Dans ce chapitre préliminaire, nous rappelons les définitions élémentaires utilisées dans le domaine de la réécriture, comme on peut les trouver dans l'article de synthèse de Dershowitz et Jouannaud [30].

1.1 Notations

Nous utilisons les notations suivantes :

- $\mathbb{N}$: ensemble des entiers naturels;
- $\mathbb{Z}$: ensemble des entiers relatifs;
- $\mathfrak{S}_n$: ensemble des permutations sur $\{1, \ldots, n\}$;

ainsi que les notations standard de la réécriture telles qu'elles ont été données par Dershowitz et Jouannaud [29, 30], que nous donnerons au fur et à mesure que nous les définirons.

1.2 Relations noethériennes et propriété de Church-Rosser

Dans cette section nous donnons des définitions générales sur les relations sur un ensemble.

Définition 1.1 *Si $\rightarrow$ est une relation binaire sur un ensemble $\mathcal{E}$, on dit que $\rightarrow$ est :*

- *réflexive si* $\forall x \in \mathcal{E}, x \rightarrow x$;
- *antiréflexive si* $\forall x \in \mathcal{E}, x \not\rightarrow x$;
- *symétrique si* $\forall x, y \in \mathcal{E}, x \rightarrow y \Longrightarrow y \rightarrow x$;
- *antisymétrique si* $\forall x, y \in \mathcal{E}, x \rightarrow y \wedge y \rightarrow x \Longrightarrow x = y$;
- *transitive si* $\forall x, y, z \in \mathcal{E}, x \rightarrow y \wedge y \rightarrow z \Longrightarrow x \rightarrow z$;
- *une relation d'équivalence si elle est à la fois réflexive, symétrique et transitive.*

Pour une relation $\rightarrow$ *quelconque, on note respectivement* $\xrightarrow{+}$, $\xrightarrow{*}$, $\longleftrightarrow$ *ses clôtures transitive, réflexive-transitive et symétrique, définies par :*

- $s \xrightarrow{+} t$ *si* $\exists s_0 = s \rightarrow s_1 \rightarrow \cdots \rightarrow s_n = t$ *avec* $n \geq 1$*;*

- $s \xrightarrow{*} t$ *si* $\exists s_0 = s \rightarrow s_1 \rightarrow \cdots \rightarrow s_n = t$ *avec* $n \geq 0$*;*

- $s \longleftrightarrow t$ *si* $s \rightarrow t$ *ou* $t \rightarrow s$*.*

Un préordre *est une relation binaire réflexive et transitive, un* ordre *est un préordre antisymétrique, et un* ordre strict *est une relation antiréflexive, antisymétrique et transitive.*

Si $\succeq$ *est un préordre, la relation* $\succ = \succeq - \preceq$ *(c'est-à-dire* $s \succ t$ *si* $s \succeq t$ *et* $t \not\preceq s$*) est un ordre strict, dit* ordre strict associé *à* $\succeq$*. La relation* $\simeq = \succeq \cap \preceq$ *(c'est-à-dire* $s \simeq t$ *si* $s \succeq t$ *et* $t \succeq s$*) est une relation d'équivalence dite* relation d'équivalence associée *à* $\succeq$*.*

Réciproquement, si $\succ$ *est un ordre strict et* $\simeq$ *une relation d'équivalence compatible avec* $\succ$ *dans le sens où* $s \simeq s' \succ t' \simeq t$ *implique* $s \succ t$*, alors* $\simeq \cup \succ$ *est un préordre. Remarquons que l'égalité est toujours compatible avec n'importe quel ordre strict* $\succ$*, et que* $= \cup \succ$ *est alors un ordre.*

1.2.1 Extension multi-ensemble

Rappelons d'abord la notion de multi-ensemble. Un multi-ensemble est une collection où les éléments peuvent apparaître plusieurs fois.

Définition 1.2 *Soit* $\mathcal{E}$ *un ensemble. Un* multi-ensemble *est une application* M *de* $\mathcal{E}$ *dans* $\mathbb{N}$ *à support fini, c'est-à-dire* $\{x \in \mathcal{E} | M(x) \neq 0\}$ *est fini. On appelle* $M(x)$ *la multiplicité de* x *dans* M*. On note les multi-ensembles en extension en énumérant* $M(x)$ *fois chaque* x*, par exemple on note* $\{a, b, b\}$ *le multi-ensemble sur* $\{a, b, c\}$ *défini par* $M(a) = 1$*,* $M(b) = 2$ *et* $M(c) = 0$*. On note* $\mathcal{M}(\mathcal{E})$ *l'ensemble des multi-ensembles d'éléments de* $\mathcal{E}$*. On définit sur* $\mathcal{M}(\mathcal{E})$ *des opérations d'appartenance, d'inclusion, d'union et de différence analogues à celles des ensembles de la manière suivante :*

- $x \in M$ *si* $M(x) \geq 1$*;*

- $M \subseteq N$ *si* $\forall x, M(x) \leq N(x)$*;*

- $(M \cup N)(x) = M(x) + N(x)$*;*

- $(M - N)(x) = Sup(0, M(x) - N(x))$*.*

Définition 1.3 *On appelle* extension multi-ensemble *d'une relation* $\rightarrow$ *sur* $\mathcal{E}$ *la clôture transitive de la relation* $\rightarrow_{mul}$ *définie sur* $\mathcal{M}(\mathcal{E})$ *par :*

$$\{s_1, \ldots, s_i, \ldots, s_n\} \rightarrow_{mul} \{s_1, \ldots, s_{i-1}, t_1, \ldots, t_k, s_{i+1}, \ldots, s_n\}$$

si $k \in \mathbb{N}$ *et* $\forall j \in \{1, \ldots, k\},\ s_i \rightarrow t_j$*.*

1.2.2 Relations noethériennes

Définition 1.4 *Une relation $\rightarrow$ sur un ensemble $\mathcal{E}$ est* noethérienne *s'il n'existe pas de chaîne infinie d'éléments en relation :*

$$\nexists (x_n)_{n\in\mathbb{N}}\ x_0 \ \rightarrow\ x_1\ \rightarrow\ x_2\ \rightarrow\ \cdots$$

Une proposition fondamentale sur les multi-ensembles est la suivante [31].

Proposition 1.5 (Dershowitz & Manna 1979) *Si $\rightarrow$ est noethérienne, alors $\rightarrow_{mul}$ est noethérienne.*

1.2.3 Propriété de Church-Rosser et confluence

On considère toujours un ensemble quelconque $\mathcal{E}$.

Définition 1.6 *Une relation $\underset{R}{\longrightarrow}$ est dite* de Church-Rosser *si*

$$\forall s,t\ s\ \overset{*}{\underset{R}{\longleftrightarrow}}\ t \Longrightarrow \exists u \mid s\ \overset{*}{\underset{R}{\longrightarrow}}\ u\ \overset{*}{\underset{R}{\longleftarrow}}\ t$$

Une relation $\underset{R}{\longrightarrow}$ est confluente *si*

$$s\ \overset{*}{\underset{R}{\longleftarrow}}\ t\ \overset{*}{\underset{R}{\longrightarrow}}\ u \Longrightarrow \exists v \mid s\ \overset{*}{\underset{R}{\longrightarrow}}\ v\ \overset{*}{\underset{R}{\longleftarrow}}\ u.$$

Une relation $\underset{R}{\longrightarrow}$ est localement confluente *si*

$$s\ \underset{R}{\longleftarrow}\ t\ \underset{R}{\longrightarrow}\ u \Longrightarrow \exists v \mid s\ \overset{*}{\underset{R}{\longrightarrow}}\ v\ \overset{*}{\underset{R}{\longleftarrow}}\ u$$

Le résultat fondamental lié à ces définitions est le lemme suivant dû à Newman [41, 74].

Lemme 1.7 *Si $\underset{R}{\longrightarrow}$ est noethérienne, alors les propriétés suivantes sont équivalentes :*

1. *$\underset{R}{\longrightarrow}$ est de Church-Rosser;*
2. *$\underset{R}{\longrightarrow}$ est confluente;*
3. *$\underset{R}{\longrightarrow}$ est localement confluente.*

On étend maintenant ces définitions au cas *modulo* : on se fixe une relation $\underset{S}{\longrightarrow}$ arbitraire sur $\mathcal{E}$.

Définition 1.8 *Soit $\underset{R}{\longrightarrow}$ une relation sur $\mathcal{E}$, on définit la* relation engendrée par $\underset{R}{\longrightarrow}$ sur les classes modulo $\overset{*}{\underset{S}{\longleftrightarrow}}$*, notée $\underset{R/S}{\longrightarrow}$ par :*

$$s\ \underset{R/S}{\longrightarrow}\ t \Longleftrightarrow \exists s', t',\ s\ \overset{*}{\underset{S}{\longleftrightarrow}}\ s'\ \underset{R}{\longrightarrow}\ t'\ \overset{*}{\underset{S}{\longleftrightarrow}}\ t$$

On note $\xrightarrow[R\cup S]{}$ *la réunion de* $\xrightarrow[R]{}$ *et* $\xrightarrow[S]{}$ *.*

On dit que $\xrightarrow[R]{}$ termine modulo $\xrightarrow[S]{}$ *si* $\xrightarrow[R/S]{}$ *est noethérienne.*

Soit $\xrightarrow[R^S]{}$ *une relation arbitraire comprise entre* $\xrightarrow[R]{}$ *et* $\xrightarrow[R/S]{}$ *. On dit que s et t sont* $\xrightarrow[R^S]{}$ -joignables modulo $\xrightarrow[S]{}$ *s'il existe u et v tels que*

$$s \xrightarrow[R^S]{*} u \xleftrightarrow[S]{*} v \xleftarrow[R^S]{*} t$$

ce que l'on note $s \downarrow_{R^S} t$*.*

On dit que $\xrightarrow[R]{}$ *est* $\xrightarrow[R^S]{}$ -Church-Rosser modulo $\xrightarrow[S]{}$ *si*

$$\forall s,t \; s \xleftrightarrow[R\cup S]{*} t \Longrightarrow s \downarrow_{R^S} t$$

On dit que $\xrightarrow[R^S]{}$ *est* confluente modulo $\xrightarrow[S]{}$ *si*

$$s \xleftarrow[R^S]{*} t \xrightarrow[R^S]{*} u \Longrightarrow s \downarrow_{R^S} u$$

On dit que $\xrightarrow[R^S]{}$ *est* cohérente modulo $\xrightarrow[S]{}$ *si*

$$s \xleftarrow[R^S]{*} t \xleftrightarrow[S]{*} u \Longrightarrow s \downarrow_{R^S} u$$

On dit que $\xrightarrow[R^S]{}$ *est* localement confluente avec $\xrightarrow[R]{}$ modulo $\xrightarrow[S]{}$ *si*

$$s \xleftarrow[R^S]{} t \xrightarrow[R]{} u \Longrightarrow s \downarrow_{R^S} u$$

On dit que $\xrightarrow[R^S]{}$ *est* localement cohérente avec $\xrightarrow[R]{}$ modulo $\xrightarrow[S]{}$ *si*

$$s \xleftarrow[R^S]{} t \xrightarrow[S]{} u \Longrightarrow s \downarrow_{R^S} u$$

Enfin, on dira que x est en forme normale pour $\xrightarrow[R^S]{}$ *s'il n'existe pas de y tel que x* $\xrightarrow[R^S]{}$ *y, et on dira que y est une forme normale de x si x* $\xrightarrow[R^S]{}$ *y et y est en forme normale.*

Le théorème suivant, dit *théorème de Church-Rosser abstrait*, généralise le lemme de Newman au cas modulo, en permettant de ramener la propriété de Church-Rosser modulo aux propriétés locales de confluence et de cohérence [44].

Théorème 1.9 (Jouannaud & Kirchner 1984) *On suppose que* $\xrightarrow[R]{}$ *termine modulo* $\xrightarrow[S]{}$ *. Alors les propriétés suivantes sont équivalentes :*

1. $\xrightarrow[R]{}$ *est* $\xrightarrow[R^S]{}$ *-Church-Rosser modulo* $\xrightarrow[S]{}$ *;*

2. $\xrightarrow[R^S]{}$ *est confluente et cohérente modulo* $\xrightarrow[S]{}$ *;*

3. $\xrightarrow[R^S]{}$ *est localement confluente et localement cohérente avec* $\xrightarrow[R]{}$ *modulo* $\xrightarrow[S]{}$ *;*

4. *pour tout* s *et* t, $s \xleftrightarrow[R \cup S]{*} t$ *si et seulement si pour toutes formes normales* s' *et* t' *de* s *et* t, $s' \xleftrightarrow[S]{*} t'$*;*

Notons que ce théorème suppose que $\xrightarrow[R/S]{}$ termine, si l'on suppose seulement que $\xrightarrow[R^S]{}$ termine il n'est plus vrai.

1.3 Algèbres universelles et termes

Dans cette section nous rappelons les définitions classiques utilisées en réécriture [30].

1.3.1 Algèbres universelles

Définition 1.10 *On appelle* signature *un couple* $(\mathcal{F}, \mathrm{ar})$ *où* $\mathcal{F}$ *est un ensemble et* ar *une fonction de* $\mathcal{F}$ *dans* $\mathbb{N}$. *On appellera* symbole de fonction *un élément de* $\mathcal{F}$ *et* arité de f *le nombre* $\mathrm{ar}(f)$.

Une $\mathcal{F}$-algèbre *est un couple* $(A, A_{\mathcal{F}})$ *où* A *est un ensemble et* $A_{\mathcal{F}}$ *une famille* $\{A_f\}_{f\in\mathcal{F}}$, *où* A_f *est une application de* $A^{\mathrm{ar}(f)}$ *dans* A.

Une application g *de* A *dans* B *est un morphisme de* $\mathcal{F}$*-algèbres si*

$$\forall f \in \mathcal{F}\ g(A_f(x_1, \ldots, x_n)) = B_f(g(x_1), \ldots, g(x_n)).$$

Exemple. Considérons la signature $\mathcal{F} = \{f, a\}$ où f est binaire (c'est-à-dire d'arité 2) et a une constante (d'arité 0). Alors $A = (\mathbb{N}, +, 0)$ et $B = (\mathbb{N}, \times, 1)$ sont des $\mathcal{F}$-algèbres, et n'importe quelle fonction exponentielle $n \mapsto c^n$ est un $\mathcal{F}$-morphisme de A dans B.

1.3.2 Algèbres de termes

Définition 1.11 *Soit* X *un ensemble disjoint de* $\mathcal{F}$ *qu'on appellera* ensemble de variables. *L'algèbre des termes sur* $\mathcal{F}$ *et* X *notée* $T(\mathcal{F}, X)$ *est le plus petit ensemble* T *vérifiant :*

- *si* $x \in X$ *alors* $x \in T$*;*
- *si* $f \in \mathcal{F}$, $\mathrm{ar}(f) = n$ *et* $t_1, \ldots, t_n \in T^n$ *alors* $f(t_1, \ldots, t_n) \in T$.

T *est une* $\mathcal{F}$*-algèbre si on le munit de* $\{T_f\}_{f\in\mathcal{F}}$ *où* T_f *est défini par:* $T_f(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$.

Proposition 1.12 $T(\mathcal{F}, X)$ *est une* $\mathcal{F}$*-algèbre* librement engendrée par X *: toute application de* X *dans une* $\mathcal{F}$*-algèbre* A *peut être prolongée de façon unique en un morphisme de* T *dans* A.

1.3.3 Positions, sous-termes et remplacement

Définition 1.13 *Une* position *dans un terme est un mot sur l'alphabet* $\mathbb{N} - \{0\}$*. Le mot vide est noté* Λ*, la concaténation de* p *et* q *est notée* $p.q$ *ou* pq*. L'ordre préfixe sur les position est défini par* $p \leq q$ *s'il existe* r *tel que* $q = p.r$*. On dit que deux positions* p *et* q *sont* parallèles, *et l'on note* $p \parallel q$*, si ni* $p \leq q$*, ni* $q \leq p$*. Si* P *est un ensemble de positions et* n *un entier, on note* $n.P$ *l'ensemble* $\{n.p \mid p \in P\}$*.*

Si s *est un terme, on appelle* domaine *de* s*, noté* $\mathcal{P}\mathrm{os}(s)$*, l'ensemble des positions de* s*, défini par :*

- *si* $x \in X$ *alors* $\mathcal{P}\mathrm{os}(x) = \{\Lambda\}$
- *si* $f \in \mathcal{F}$, $\mathrm{ar}(f) = n$ *et* $t_1, \ldots, t_n \in T^n$ *alors* $\mathcal{P}\mathrm{os}(f(t_1, \ldots, t_n)) = \{\Lambda\} \cup 1.\mathcal{P}\mathrm{os}(t_1) \cup \ldots \cup n.\mathcal{P}\mathrm{os}(t_n)$

On définit le domaine clos *de* s *noté* $\mathcal{FP}\mathrm{os}(s)$ *par:*

- *si* $x \in X$ *alors* $\mathcal{FP}\mathrm{os}(x) = \emptyset$
- $\mathcal{FP}\mathrm{os}(f(t_1, \ldots, t_n)) = \{\Lambda\} \cup 1.\mathcal{FP}\mathrm{os}(t_1) \cup \ldots \cup n.\mathcal{FP}\mathrm{os}(t_n)$

Le symbole de tête *d'un terme* s*, noté* $\mathcal{H}\mathrm{ead}(s)$ *est défini par :*

- *si* $x \in \mathcal{X}$ *alors* $\mathcal{H}\mathrm{ead}(x) = x$*;*
- $\mathcal{H}\mathrm{ead}(f(t_1, \ldots, t_n)) = f$*.*

L'ensemble des variables de s*, noté* $\mathcal{V}\mathrm{ar}(s)$*, est défini par :*

- *si* $x \in X$ *alors* $\mathcal{V}\mathrm{ar}(x) = \{x\}$*;*
- $\mathcal{V}\mathrm{ar}(f(t_1, \ldots, t_n)) = \mathcal{V}\mathrm{ar}(t_1) \cup \ldots \cup \mathcal{V}\mathrm{ar}(t_n)$*.*

Si $s \in T$ *et* $p \in \mathcal{P}\mathrm{os}(s)$*, on définit le* sous-terme de s à la position p*, noté* $s|_p$*, par:*

- $s|_\Lambda = s$
- $f(t_1, \ldots, t_n)|_{ip} = t_i|_p$

Enfin, on définit le remplacement de s par t à la position p*, noté* $s[t]_p$*, par:*

- $s[t]_\Lambda = t$
- $f(t_1, \ldots, t_n)[t]_{ip} = f(t_1, \ldots, t_i[t]_p, \ldots, t_n)$

Nous intéressons particulièrement au cas les termes que l'on manipule sont sans variables. Nous sommes ainsi amené à la définition suivante.

Définition 1.14 *On dira qu'un terme* s *est* clos *lorsque* $\mathcal{V}\mathrm{ar}(s) = \emptyset$*.*

Pour terminer les définitions liées aux positions et aux sous-termes, nous donnons la définition de l'ordre de sous-terme. En effet, la propriété sous-terme définie ci-dessus engendre en fait une relation qui est trivialement une relation d'ordre :

Définition 1.15 L'ordre sous-terme *est la relation* $\trianglerighteq$ *sur* $T(\mathcal{F}, X)$ *définie par* $s \trianglerighteq t$ *s'il existe* $p \in \mathcal{P}\mathrm{os}(s)$ *tel que* $t = s|_p$*.*

1.3.4 Substitutions, filtrage et unification

Définition 1.16 *On appelle* substitution *un endomorphisme de $T(\mathcal{F}, X)$. On note $s\sigma$ le résultat de l'application de σ à un terme s et $\theta\sigma$ la composition de σ suivie de θ (ainsi $(s\theta)\sigma = s(\theta\sigma)$). L'algèbre T étant librement engendrée par X, une substitution est caractérisée par les images des variables. Lorsque toutes les variables sauf un nombre fini sont substituées à elle-même, on notera une substitution sous la forme $\{x_1 \mapsto s_1, \ldots, x_n \mapsto s_n\}$*

Une substitution bijective est un appelée un renommage, *auquel cas l'image de toute variable est une variable, et l'application ainsi induite sur X est une bijection.*

Définition 1.17 *On dit qu'un terme t* filtre *un terme s, ou encore que s est une* instance *de t, s'il existe une substitution σ telle que $s = t\sigma$, ce qu'on note $s \geq t$. La relation $\geq$ est une relation d'ordre appelée* ordre de subsumption. *On étend cette définition aux substitutions : $\sigma \geq \theta$ s'il existe une substitution λ telle que $\sigma = \theta\lambda$.*

Définition 1.18 *L'ordre de spécialisation est la relation $\trianglerighteq$ définie par $t \trianglerighteq t$ s'il existe une position $p \in \mathcal{P}\mathrm{os}(s)$ et une substitution σ tels que $s|_p = t\sigma$.*

Proposition 1.19 *Les relations d'équivalence associées à $\geq$ et à $\trianglerighteq$ sont toutes les deux l'égalité modulo renommage.*

Un certain nombre de définitions sur les termes s'étendent de manière canonique aux substitutions. Nous avons en particulier les définitions suivantes.

Définition 1.20 *Une propriété sur les termes s'étend aux substitutions en disant que σ vérifie cette propriété si $x\sigma$ vérifie la propriété, pour tout x ou pour au moins un x suivant les cas. Ainsi :*

- *une substitution σ est dite* close *si pour toute variable x du domaine de σ, $x\sigma$ est clos (c'est-à-dire $x\sigma$ clos ou $x\sigma = x$);*
- *si $\longrightarrow$ est une relation arbitraire sur l'algèbre des termes, $\longrightarrow$ s'étend aux substitutions par la définition $\sigma \longrightarrow \sigma'$ si pour un certain x, $x\sigma \longrightarrow x\sigma'$ et $y\sigma = y\sigma'$ pour tout y différent de x, et on vérifie qu'avec cette définition, σ est en forme normale pour $\longrightarrow$ si pour tout x, $x\sigma$ l'est.*

Une opération fondamentale qui va nous servir continuellement est l'unification, que nous définissons maintenant.

Définition 1.21 *Deux termes s et t sont* unifiables *s'il existe une substitution σ telle que $s\sigma = t\sigma$. Un tel σ s'appelle un* unificateur *de s et t. θ est un* unificateur principal *(ou* unificateur plus général*) de s et t s'il est maximal pour $\geq$, c'est-à-dire s'il n'existe pas d'unificateur σ tel que $\theta > \sigma$.*

Proposition 1.22 *l'unification est décidable et si deux termes s et t sont unifiables, ils admettent des unificateurs principaux qui sont tous égaux à renommage près, on note un tel unificateur $mgu(s, t)$ [38, 83] (pour l'équivalent anglophone* most general unifier*).*

1.3.5 Théories équationnelles

Syntaxe

Définition 1.23 *Une* équation *est une paire de termes $\{s,t\}$, notée $s = t$. On dira qu'une équation $s = t$ est* close *lorsque s et t sont clos.*

Définition 1.24 *Une relation binaire $\rightarrow$ sur $T(\mathcal{F},X)$ est dite* stable par mise sous contexte *si pour tous s, t, u et pour tout $p \in \mathcal{P}os(u)$*

$$s \rightarrow t \Longrightarrow u[s]_p \rightarrow u[t]_p$$

et elle est dite stable par instanciation *si pour tous s, t et σ*

$$s \rightarrow t \Longrightarrow s\sigma \rightarrow t\sigma.$$

Si elle vérifie ces deux propriétés, elle est dite terme-compatible.

Définition 1.25 *Une* théorie équationnelle *est une relation d'équivalence terme-compatible sur $T(\mathcal{F},X)$.*

Définition 1.26 *Si E est un ensemble d'équations, on note $\underset{E}{\longleftrightarrow}$ la plus petite relation sur $T(\mathcal{F},\mathcal{X})$ symétrique terme-compatible qui contienne E. Ainsi :*

$$s \underset{E}{\longleftrightarrow} t \Longleftrightarrow \exists \sigma, u, p, \begin{cases} s = u[a\sigma]_p \\ t = u[b\sigma]_p \end{cases} \quad \text{où } a = b \in E$$

La clôture réflexive-transitive $\overset{*}{\underset{E}{\longleftrightarrow}}$ est alors une théorie équationnelle dite théorie équationnelle engendrée par E. *On pourra dire « la théorie équationnelle E » par abus de langage.*

Sémantique

Définition 1.27 *Une équation $s = t$ est* valide *dans une $\mathcal{F}$-algèbre A si pour tout morphisme g de $T(\mathcal{F},X)$ dans A on a $g(s) = g(t)$. Cela signifie que s et t dénotent le même élément de A pour toute interprétation possible des variables de s et t*

Si E est un ensemble d'équations, E est valide *dans A si chaque équation de E l'est, on note alors $A \models E$, et on dira également que A est un* modèle *de E.*

Une équation $s = t$ est une conséquence de E si $s = t$ est valide dans tout modèle de E (c'est-à-dire dans toute algèbre qui valide E), ce que l'on note $E \models s = t$. La relation $=_E$ définie par $s =_E t$ si $E \models s = t$ est une relation d'équivalence appelée l'égalité modulo E.

Exemple. Considérons la signature $\mathcal{F} = \{0, s, +\}$ où 0 est une constante, s est unaire et $+$ binaire (noté de façon infixe) et considérons la $\mathcal{F}$-algèbre $\mathbb{N}$ où les opérations sont respectivement l'entier 0, la fonction successeur et l'addition habituelle. Alors l'équation de commutativité $x+y = y+x$ est valide dans cette algèbre. Par contre, elle n'est pas conséquence des équations $x + 0 = x$ et $x + s(y) = s(x + y)$: en effet on a par exemple l'algèbre définie par $A = \mathbb{N}^2$, $A_0 = (0,0)$, $A_s(n,m) = (n+1,m)$ et $A_+((n,m),(n',m')) = (n+n', m+2m')$ dans laquelle la commutativité n'est pas valide, alors que les deux équations précédentes le sont.

Théorème 1.28 (Birkhoff 1935) *$E \models s = t$ si et seulement si $s \xleftrightarrow[E]{*} t$.*

Grâce à ce théorème, nous pouvons à partir de maintenant confondre sans danger la notation syntaxique d'égalité et la notion sémantique de validité.

Exemple. Les termes $x + y$ et $y + x$ ne sont pas égaux dans la théorie équationnelle engendrée par $\{x + 0 = x, x + s(y) = s(x + y)\}$ puisque ce n'est pas vrai dans le modèle de l'exemple précédent. Autrement dit, on ne peut pas prouver la commutativité avec seulement des remplacement d'égaux avec les équations $x + 0 = x$ et $x + s(y) = s(x + y)$.

Définition 1.29 *L'algèbre* quotient *$A/_E$ est le quotient de A par la relation $=_E$, c'est encore une $\mathcal{F}$-algèbre pour les opérations quotientées.*

Problèmes du mot

Définition 1.30 *Le* problème du mot *pour une théorie E est le problème suivant :*

Donnée	:	*deux termes s et t*
Question	:	*est-ce que $s =_E t$?*

Le problème du mot uniforme *pour une classe de théories équationnelles $\mathcal{C}$ est le problème suivant :*

Donnée	:	*une théorie E de $\mathcal{C}$ et deux termes s et t*
Question	:	*est-ce que $s =_E t$?*

Il est connu que le problème du mot uniforme pour toutes les théories équationnelles est indécidable. Nous nous intéresserons au chapitre 5 au problème du mot uniforme pour certaines classes de théories : les théories E-closes pour certains E particuliers.

Définition 1.31 *Soit E un ensemble d'équations. Une* théorie close modulo E, *ou E-close, est une théorie équationnelle définie par la réunion de E et d'un ensemble arbitraire d'équations closes.*

On peut remarquer qu'avec cette définition, n'importe quelle théorie équationnelle est une théorie E-close pour un certain E. Ce qui nous intéresse, c'est d'étudier la classe des théories E-closes pour un certain nombre de théories E particulières : on s'intéressera aux théories engendrées par les axiomes classiques d'associativité, de commutativité, d'existence d'un élément neutre, d'inverse, d'idempotence, de nilpotence, de distributivité, etc. Ces axiomes, avec leurs notations, sont énumérés dans la figure 1.1.

1.3.6 Filtrage et unification modulo une théorie équationnelle

Soit E une théorie équationnelle.

Définition 1.32 *On dit qu'un terme t* filtre modulo E *un terme s s'il existe une substitution σ telle que $s =_E t\sigma$, ce qu'on note $s \geq_E t$, ce qui définit* l'ordre de subsumption modulo E.

Contrairement au cas standard, l'équivalence associée à $\geq_E$ n'est pas nécessairement l'égalité modulo renommage et E : c'est faux dès que E est tel qu'un terme est égal à l'une de ses instances, par exemple si $E = \{f(x) = f(a)\}$.

Nom	Notation	Équations
Associativité	$A(+)$	$(x+y)+z = x+(y+z)$
Commutativité	$C(+)$	$x+y = y+x$
Associativité et commutativité	$AC(+)$	$A(+) \cup C(+)$
élément neutre à droite	$1_d(+,0)$	$x+0 = x$
élément neutre à gauche	$1_g(+,0)$	$0+x = x$
élément neutre	$1(+,0)$	$1_d(+,0)$ et $1_g(+,0)$
AC et élément neutre	$AC1(+,0)$	$AC(+)$ et $x+0=x$
Idempotence	$I(+)$	$x+x = x$
AC et idempotence	$ACI(+)$	$AC(+)$ et $I(+)$
Élément absorbant	$AC0(.,0)$	$AC(.)$ et $x.0 = 0$
Inverse à droite		$x+(-x) = 0$
Inverse à gauche		$(-x)+x = 0$
Théorie des Groupes	$G(.,e,I)$	$A(.)$, $1(.,e)$ et $x.I(x) = e$
k-nilpotence de +	$N_k(+,0)$	$AC(+)$ et $\underbrace{x+\cdots+x}_{k \text{ fois}} = 0$
Nilpotence de +	$N(+,0)$	$N_2(+,0)$
Distributivité gauche de $*$ par rapport à +	$D_g(*,+)$	$(x+y)*z = (x*z)+(y*z)$
Distributivité droite de $*$ par rapport à +	$D_d(*,+)$	$x*(y+z) = (x*y)+(x*z)$
Distributivité de $*$ par rapport à +	$D(*,+)$	$D_g(*,+)$ et $D_d(*,+)$
Distributivité AC	$ACD(*,+)$	$AC(*)$, $AC(+)$ et $D(*,+)$
Théorie des Groupes abéliens	$AG(+,0,-)$	$AC(+)$, $1(+,0)$ et $x+(-x) = 0$
Anneaux commutatifs	$CR(+,0,-,.,1)$	$AG(+,0,-)$, $ACD(.,+)$ et $AC1(.,1)$

Figure 1.1 : Axiomes équationnels classiques

Définition 1.33 *Une théorie E est* régulière *si $s =_E t$ implique $\mathcal{V}\mathrm{ar}(s) = \mathcal{V}\mathrm{ar}(t)$.*

Proposition 1.34 *Si E est régulière, l'équivalence associée à $\geq_E$ est l'égalité modulo renommage et E.*

La théorie régulière qui nous intéresse principalement est AC. Les algorithmes de filtrage AC sont maintenant bien connus, ce problème est NP-complet [53], mais beaucoup de chercheurs travaillent pour obtenir quand même des algorithmes efficaces, en particulier on cherche à *compiler* le filtrage par un ensemble de termes donnés, afin d'accélérer le processus de réécriture AC.

Définition 1.35 *Deux termes s et t sont* unifiables modulo E *s'il existe une substitution σ telle que $s\sigma =_E t\sigma$. Un tel σ s'appelle un E-*unificateur *de s et t. θ est un E-unificateur principal de s et t s'il est maximal pour $\geq_E$ autrement dit il n'existe pas de E-unificateur σ tel que $\theta >_E \sigma$. Un ensemble de substitutions Σ est un ensemble complet de E-unificateurs de s et t si ce sont des E-unificateurs et que tout E-unificateur est E-instance de l'un d'eux. C'est*

Théorie	Décidabilité	Type
$\emptyset$	Oui	unitaire
$A(+)$	Oui	infinitaire
$C(+)$	Oui	finitaire
$AC(+)$	Oui	finitaire
$AC1(+,0)$	Oui	unitaire
$ACI(+)$	Oui	unitaire
$ACD(*,+)$	Non	infinitaire
$AG(+,0,-)$	Oui	finitaire
$BR(+,0,-,.,1)$	Oui	unitaire

Figure 1.2 : Classification de quelques théories équationnelles vis-à-vis de l'unification

un ensemble minimal *complet d'unificateurs s'ils sont deux à deux incomparables pour* $\geq_E$*, et on note un tel ensemble* $CSU_E(s,t)$ *(pour l'équivalent anglophone* Complete Set of Unifiers*).*

La E-unifiabilité n'est pas décidable en général, et l'on peut classifier les théories équationnelles en fonction de leur comportement vis-à-vis de l'unification (décidable ou indécidable) et de la taille des CSU [54] :

- unitaire : il existe toujours un CSU à un élément;
- finitaire : il existe toujours un CSU fini;
- infinitaire : il existe toujours un CSU, mais éventuellement infini;
- nullaire : il peut ne pas exister de CSU.

Ces résultats sur la E-unification ne suffisent encore pas pour notre propos, car nous allons unifier des termes contenant des symboles libres, c'est-à-dire des symboles n'apparaissant pas dans E. Par exemple, on peut chercher à unifier modulo AC les termes $x+(-x)$ et $y+0$, et dans ce problème $-$ et 0 sont des symboles libres. La présence de tels symboles change le type de l'unification, par exemple l'unification AC1, qui est unitaire, devient finitaire en présence de symboles libres.

Un autre problème apparaît lorsque l'on se trouve en présence en même temps de symboles satisfaisant des théories différentes, par exemple on peut avoir envie d'unifier des termes contenant un symbole AC et un symbole ACI. La question est alors de savoir si l'on peut combiner les algorithmes d'unification modulo AC et ACI correspondant, c'est le problème du *mélange* de théories. Pour une étude détaillée du problème de l'unification dans un mélange de théories, on pourra se référer à divers articles sur l'unification [16, 22, 34, 54]. Nous nous bornons ici à rappeler les résultats qui nous intéressent :

- l'unification en présence de symboles libres et d'un nombre arbitraire de symboles AC est décidable et finitaire, le nombre de solutions d'une équation est une fonction doublement exponentielle de la taille des termes à unifier;
- l'unification en présence de symboles libres et de symboles AC, soit munis d'éléments neutres (AC1), soit idempotent (ACI), soit les deux (AC1I), est décidable et finitaire,

le nombre de solutions d'une équation est une fonction simplement exponentielle de la taille des termes à unifier;

- l'unification modulo les théorie des groupes abéliens en présence de symboles libres est finitaire, ainsi que modulo la théorie des anneaux booléens [18].

1.3.7 Relations de réécriture

La notion de réécriture se définit de manière analogue à l'égalité modulo un ensemble d'équations, mais où les équations sont remplacées par des règles, c'est-à-dire sont utilisées uniquement dans un sens.

Définition 1.36 *Une* règle *est un couple de termes (s, t), noté $s \to t$.*

Soit R un ensemble de règles.

Définition 1.37 *La* relation de réécriture *par R, notée $\underset{R}{\longrightarrow}$, est la plus petite relation terme-compatible qui contienne R. Autrement dit :*

$$s \underset{R}{\longrightarrow} t \iff \exists \sigma, u, p, \begin{cases} s = u[l\sigma]_p \\ t = u[r\sigma]_p \end{cases} \quad \textit{où } l = r \in R$$

Définition 1.38 *On dira qu'un terme s est R-*réductible *s'il existe t tel que $s \underset{R}{\longrightarrow} t$, sinon on dira que s est R-*irréductible, *ou encore est en* forme normale *pour R. On dira que t est une R-*forme normale *de s si $s \overset{*}{\underset{R}{\longrightarrow}} t$ et t est R-irréductible. On note alors $t = s\downarrow_R$. On étend naturellement cette définition aux substitutions : $x(\sigma\downarrow_R) = (x\sigma)\downarrow_R$.*

Soit E une théorie équationnelle.

Définition 1.39 *La relation de réécriture par R* modulo *E, notée $\underset{R/E}{\longrightarrow}$ est la plus petite relation contenant $\underset{R}{\longrightarrow}$ et compatible avec E, autrement dit :*

$$s \underset{R/E}{\longrightarrow} t \iff \exists s', t', \begin{cases} s =_E s' \\ s' \underset{R}{\longrightarrow} t' \\ t' =_E t \end{cases}$$

On peut interpréter également la réécriture modulo E comme la réécriture sur les classes modulo E.

La réécriture modulo E est une relation très générale, car pour réécrire on est plus ou moins obligé de parcourir sa classe modulo E. Pour éviter cette combinatoire, on est amené à considérer une relation intermédiaire entre $\underset{R}{\longrightarrow}$ et $\underset{R/E}{\longrightarrow}$ qui s'appelle la réécriture E-étendue.

Définition 1.40 *La relation de réécriture E-*étendue *par R, notée $\underset{E\backslash R}{\longrightarrow}$ est définie par :*

$$s \overset{p,\sigma}{\underset{E\backslash R}{\longrightarrow}} t \iff \begin{cases} l \to r \in R \\ s|_p =_E l\sigma \\ t = s[r\sigma]_p \end{cases}$$

Exemple. Le terme $a + (b + c)$ se réécrit modulo AC par la règle $a + b \to d$ en $d + c$, mais ne se réécrit pas de façon AC-étendue, car l'étape d'égalité modulo AC à faire avant d'appliquer la règle est à une position au-dessus de la position à laquelle on applique la règle. On voit bien sur cet exemple qu'il faut parcourir la classe modulo AC de $a + (b + c)$ pour le réécrire modulo AC.

Définition 1.41 *On dira qu'un système de règles R* termine, *ou* est à terminaison finie *lorsque la relation $\underset{R}{\longrightarrow}$ est noethérienne. On dira que R est* confluent *si la relation $\underset{R}{\longrightarrow}$ est confluente. On dira que R est* convergent *s'il est à la fois à terminaison finie et est confluent. On a les définitions analogues pour le cas modulo.*

Ces propriétés sont liées à l'existence et à l'unicité des formes normales d'un terme, comme l'indique le lemme suivant.

Lemme 1.42 *Si R termine, tout terme possède au moins une forme normale, et si R est confluent, tout terme possède au plus une forme normale.*

Le théorème de Church-Rosser abstrait, appliqué à la réécriture étendue, qui est bien intermédiaire entre la réécriture standard et la réécriture modulo, nous permet d'obtenir la proposition suivante.

Proposition 1.43 *Si R termine modulo E, alors la réécriture étendue $E\backslash R$ est convergente si et seulement elle est localement confluente et localement cohérente avec E.*

Une propriété également importante des systèmes de réécriture est l'inter-réduction, on verra en particulier des résultats d'unicité liés à cette propriété.

Définition 1.44 *On dira que R est* inter-réduit *si pour toute règle $l \to r$ de R, r est R-irréductible, et tout terme inférieur à l pour l'ordre de spécialisation est R-irréductible. On dira enfin que R est* canonique *s'il est à la fois convergent et inter-réduit.*

Dans le cas d'un système convergent, on a une définition équivalente de l'inter-réduction, donnée par la proposition suivante.

Proposition 1.45 *Si R est convergent, R est inter-réduit si et seulement si pour toute règle $l \to r$ de R, r est R-irréductible et l est $R - \{l \to r\}$-irréductible.*

Démonstration. Si un terme inférieur à l pour l'ordre de spécialisation est R-réductible, alors l l'est aussi, et ce ne peut pas être par la règle $l \to r$ elle-même. Réciproquement, si l est $R - \{l \to r\}$-réductible, si ce n'est pas en tête ou si c'est avec une substitution qui n'est pas un renommage, on conclut aisément, et si c'est en tête par un renommage, on a une règle $l \to r'$ dans R, mais comme R est convergent, cela impose $r = r'$. □

1.3.8 Algorithmes de complétion

L'idée de base que l'on va essayer de mettre en œuvre dans la suite est d'utiliser la réécriture pour résoudre le problème du mot d'une théorie équationnelle donnée. Plus précisément, étant donnée une théorie définie par un ensemble d'équations E, on cherche à construire un ensemble de règles convergent dont la théorie équationnelle associée soit la même que celle de E. On aura alors une procédure de décision grâce à la propriété : $s =_E t$ si et seulement si $s \downarrow_R = r \downarrow_R$. C'est ce procédé de construction d'un système R à partir de E qui s'appelle *complétion.*

Nous allons nous intéresser à montrer la décidabilité du problème du mot uniforme sur les théories E-closes pour certains E, et nous chercherons dans ce cas à utiliser la réécriture modulo E.

Nous présenterons les algorithmes de complétion par des ensembles de règles d'inférences, ce qui a l'avantage de bien séparer les *inférences* de la *stratégie* d'application des inférences.

Définition 1.46 *Une* règle d'inférence *est une transformation $E; R \vdash E'; R'$ où E et E' sont des ensembles d'équations et R et R' sont des ensembles de règles de réécriture.*

Un algorithme de complétion *associé à un ensemble de règles d'inférence donné est un algorithme qui prend en entrée un ensemble d'équations E_0 et qui produit une suite finie ou infinie $(E_n; R_n)$ où $R_0 = \emptyset$ et pour tout i, $E_i; R_i \vdash E_{i+1}; R_{i+1}$ pour l'une des règles d'inférence. Dans le cas où cette suite est finie, s'arrêtant à $E_n; R_n$, on pose $E_i = E_n$ et $R_i = R_n$ pour $i \geq n$, et on pose alors :*

$$E_\infty = \bigcup_{n=0}^{\infty} \left(\bigcap_{i=n}^{\infty} E_i \right) \ , \ R_\infty = \bigcup_{n=0}^{\infty} \left(\bigcap_{i=n}^{\infty} R_i \right)$$

Les ensembles E_∞ et R_∞ sont respectivement l'ensemble des équations persistantes et l'ensemble des règles persistantes. On dira que la complétion échoue *si E_∞ est non vide et* réussit *sinon. On dira que la complétion* diverge *si la suite est infinie.*

Il faut bien remarquer qu'avec cette définition, la complétion peut réussir et diverger et même temps, et que d'autre part la complétion peut diverger bien que R_∞ soit fini (nous verrons un tel exemple au chapitre 5).

1.3.9 Algèbre des termes variadiques

Nous définissons maintenant *l'algèbre des termes variadiques.* Il s'agit d'une généralisation de la notion d'algèbre de termes de la section précédente, où certains symboles de fonction ont une arité non fixée. Cette généralisation va nous permettre de coder les axiomes d'associativité et de commutativité dans la structure des termes, par le biais de l'opération *d'aplatissement.*

Définition 1.47 *Une* signature variadique *est une signature où la fonction d'arité n'est pas définie partout. Les symboles dont l'arité n'est pas définie s'appellent les* symboles variadiques. *L'algèbre des termes variadiques $TV(\mathcal{F}, \mathcal{FV}, \mathcal{X})$ sur la signature $\mathcal{F}$ et sur l'ensemble de symboles variadiques $\mathcal{FV}$ est définie par :*

- *si $f \in \mathcal{F}$, $\mathrm{ar}(f) = n \geq 0$, et $t_1, \ldots, t_n \in TV$ alors $f(t_1, \ldots, t_n) \in TV$.*
- *si $f \in \mathcal{FV}$, $n \geq 0$, et $t_1, \ldots, t_n \in TV$ alors $f(t_1, \ldots, t_n) \in TV$.*

Dans la suite, nous utiliserons comme ensemble de symboles variadiques l'ensemble des symboles AC. C'est pourquoi nous allons noter dorénavant les symboles variadiques de façon infixe, et habituellement par un symbole d'opération classique comme $+$.

L'*aplatissement* consiste à transformer un terme de manière à coder l'associativité et la commutativité dans la structure du terme. Par exemple, $(a+b)+c$ est transformé en $a+b+c$.

Définition 1.48 L'aplatissement *est une application notée* flat *de* TV *dans* TV :

- *si* $f \in \mathcal{F}$, $\mathrm{flat}(f(t_1,\ldots,t_n)) = f(\mathrm{flat}(t_1),\ldots,\mathrm{flat}(t_n))$;
- *si* $+ \in \mathcal{FV}$, $\mathrm{flat}(s_1+\cdots+s_n) = t_{1,1}+\cdots+t_{1,k_1}+\cdots+t_{n,1}+\cdots+t_{n,k_n}$ *où :*
  - *si* $\mathcal{H}\mathrm{ead}(s_i) \neq +$ *alors* $k_i = 1$ *et* $t_{i,1} = s_i$,
  - *si* $\mathcal{H}\mathrm{ead}(s_i) = +$ *alors* $t_{i_1}+\cdots+t_{i,k_i} = \mathrm{flat}(s_i)$.

Un terme s *sera dit* aplati *si* $s = \mathrm{flat}(s)$.

Nous noterons aussi parfois $\mathrm{flat}(s)$ par $\overline{s}$.

L'associativité et la commutativité correspondent à la *congruence de permutation* sur les termes aplatis, qui est définie de la façon suivante :

Définition 1.49 *pour* s, t *dans* TV :

$$s \equiv t \Longleftrightarrow \begin{cases} s = f(s_1,\ldots,s_n) \\ t = f(t_1,\ldots,t_n) \\ \forall i,\ s_i \equiv t_i \end{cases} \quad ou \quad \begin{cases} s = s_1+\cdots+s_n \\ t = t_1+\cdots+t_m \\ m = n \\ \exists \tau \in \mathfrak{S}_n,\ \forall i,\ s_i \equiv t_{\tau(i)} \end{cases}$$

Lemme 1.50 *Si* $\mathcal{F}_{\mathrm{AC}} = \mathcal{FV}$, *alors pour tout* s, t *dans* T:

$$s =_{\mathrm{AC}} t \Longleftrightarrow \overline{s} \equiv \overline{t}$$

Une des difficultés posée par les termes variadiques est la représentation des positions et les sous-termes. Par exemple, quelle est la position de $b+c$ dans $a+b+c+d$? Nous allons représenter cette position par l'ensemble des indices $\{2,3\}$.

Définition 1.51 *L'ensemble des positions d'un terme variadique* s, *noté* $\overline{\mathcal{P}\mathrm{os}}(t)$, *est défini par :*

- $\overline{\mathcal{P}\mathrm{os}}(f(t_1,\ldots,t_n)) = \{\Lambda\} \cup 1.\overline{\mathcal{P}\mathrm{os}}(t_1) \cup \ldots \cup n.\overline{\mathcal{P}\mathrm{os}}(t_n)$;
- $\overline{\mathcal{P}\mathrm{os}}(t_1+\ldots+t_n) = \{\Lambda\} \cup 1.\overline{\mathcal{P}\mathrm{os}}(t_1) \cup \ldots \cup n.\overline{\mathcal{P}\mathrm{os}}(t_n) \cup \mathcal{P}_n$ *où* $\mathcal{P}_n$ *est l'ensemble des parties de* $\{1,\ldots,n\}$, *excepté l'ensemble vide et la partie pleine.*

Les positions dans $\mathcal{P}_n$ *seront encore appelées les positions de premier niveau de* t. *Plus généralement, le* niveau *d'une position aplatie est obtenu en effaçant l'ensemble se trouvant éventuellement à la fin de cette position :*

- $\mathrm{Level}(\Lambda) = \Lambda$
- $\mathrm{Level}(i.p) = i.\,\mathrm{Level}(p)$

- $\text{Level}(\{i_1, \ldots, i_k\}) = \Lambda$

À présent, le remplacement peut être étendu aux termes variadiques de la façon suivante :

Définition 1.52 *Si* $A \in \mathcal{P}_n$,

$$(t_1 + \ldots + t_n)[t]_A = \text{flat}(t_{i_1} + \cdots + t_{i_k} + t)$$

où $\{i_1, \ldots, i_k\} = \{1, \ldots, n\} - A$. *Il faut remarquer que le réaplatissement est nécessaire si* $\mathcal{H}\text{ead}(t)$ *est* $+$, *et que le remplacement est défini modulo* $\equiv$.

Exemple. $(a + b + c + d)[e + f + g]_{\{1,3\}} = b + d + e + f + g$

Quand on aplatit un terme s, l'ensemble de ses positions est transformé. Cela définit une application flat_s de $\mathcal{P}\text{os}(s)$ dans $\overline{\mathcal{P}\text{os}}(s)$. Cette application est complètement spécifiée par :

- $\text{flat}_{f(s_1,\ldots,s_n)}(\Lambda) = \Lambda$;
- $\text{flat}_{f(s_1,\ldots,s_n)}(i.p) = i.\,\text{flat}_{s_i}(p)$;

et si $\text{flat}(s) = s_1 + \ldots + s_n$ et $\text{flat}(t) = t_1 + \ldots + t_m$ $(n \geq 1, m \geq 1)$;

- $\text{flat}_{s+t}(\Lambda) = \Lambda$
- si $n = 1$ alors $\text{flat}_{s+t}(1.p) = 1.\,\text{flat}_s(p)$
- si $m = 1$ alors $\text{flat}_{s+t}(2.p) = (n + 1).\,\text{flat}_t(p)$
- si $n \geq 2$ alors $\text{flat}_{s+t}(1) = \{1, \ldots, n\}$
- si $m \geq 2$ alors $\text{flat}_{s+t}(2) = \{n + 1, \ldots, m\}$
- si $n \geq 2$ alors $\text{flat}_{s+t}(1.i.p) = \text{flat}_s(i.p)$
- si $m \geq 2$ et $\text{flat}_t(i.p) = j.q$ alors $\text{flat}_{s+t}(2.i.p) = (n + j).q$
- si $m \geq 2$ et $\text{flat}_t(i.p) = \{i_1, \ldots, i_k)$ alors $\text{flat}_{s+t}(2.i.p) = \{n + i_1, \ldots, n + i_k)$

Exemple. pour $s = (a + b) + (c * d)$ on a :

$$\begin{array}{ll} \multicolumn{2}{c}{\text{flat}_s(\Lambda) = \Lambda} \\ \text{flat}_s(1) = \{1, 2\} & \text{flat}_s(2) = 3 \\ \text{flat}_s(1.1) = 1 & \text{flat}_s(2.1) = 3.1 \\ \text{flat}_s(1.2) = 2 & \text{flat}_s(2.2) = 3.2 \end{array}$$

Lemme 1.53 *si* s, t *sont des termes et* $p \in \mathcal{P}\text{os}(s)$, $\overline{p} = \text{flat}_s(p)$:

- $\overline{s|_p} = \overline{s}|_{\overline{p}}$
- $\overline{s[t]_p} = \overline{s}[\overline{t}]_{\overline{p}}$

Démonstration. Par induction sur la structure. □

On étend à TV de façon naturelle les notions de substitution, de stabilité par mise sous contexte et par instanciation, de terme-compatibilité.

Chapitre 2

Relations d'ordre sur les termes

Les ordres sur les algèbres de termes jouent pour nous un rôle particulièrement important, car ce sont eux qui permettent de prouver la terminaison des systèmes de réécriture. Un article très complet sur les techniques de preuve de terminaison a été écrit par Dershowitz [28].

Nous allons décrire ici les ordres dont nous aurons besoin dans la suite. Nous aurons besoin en particulier d'ordres qui soient compatibles avec l'associativité et la commutativité, pour pouvoir assurer la terminaison de la réécriture modulo ces axiomes. La définition de tels ordres est loin d'être triviale et nous rappelons les définitions déjà connues.

Un problème difficile, résolu seulement récemment (en 1991) par Narendran et Rusinowitch [71], est la construction d'un ordre AC total sur les classes modulo AC. Plus récemment encore, Nieuwenhuis et Rubio [76] ont défini un autre ordre AC total, plus intéressant car basé sur une précédence entre opérateurs. La construction d'un tel ordre est essentielle pour nous car seul un ordre total nous permettra d'obtenir nos résultats de décidabilité.

Néanmoins, ni l'ordre de Narendran et Rusinowitch, ni celui de Nieuwenhuis et Rubio n'est encore suffisant pour obtenir le résultat de décidabilité voulu sur les anneaux commutatifs, et nous donnons en fin de chapitre une nouvelle généralisation de cet ordre qui nous permet en particulier de définir un ordre AC total qui oriente « correctement » les règles du système canonique classique de la théorie des anneaux, c'est-à-dire qui oriente la distributivité dans le sens standard. Un tel ordre nous permettra de faire une complétion normalisée modulo les anneaux de n'importe quelle théorie close.

2.1 Ordres bien fondés

Dans cette section nous rappelons les définitions des propriétés classiques des ordres que l'on exige en réécriture.

2.1.1 Ordres bien fondés, extensions lexicographique et multi-ensemble

Définition 2.1 *Un préordre $\succeq$ est* bien fondé *si son ordre strict associé est noethérien.*

La fonctionnelle *lexicographique* permet de construire de nouveaux ordres à partir d'anciens.

Définition 2.2 *Si $E_1, \ldots, E_n$ sont munis respectivement des préordres $\succeq_1, \ldots, \succeq_n$ alors on définit* l'extension lexicographique $\succeq = (\succeq_1, \ldots, \succeq_n)_{lex}$ *sur $E_1 \times \ldots \times E_n$ par:*

$$s \succ t \iff \exists k \in \{1, \ldots, n\} \begin{cases} \forall i < k \ s_i \succeq_i t_i \\ s_k \succ_k t_k \end{cases}$$
$$s \simeq t \iff \forall i \ s_i \simeq_i t_i$$

Proposition 2.3 *Si $\succeq_1, \ldots, \succeq_n$ sont bien-fondés alors $(\succeq_1, \ldots, \succeq_n)_{lex}$ est bien-fondé.*

La fonctionnelle *multi-ensemble*, déjà définie au chapitre 1 pour une relation quelconque, s'applique en particulier aux relations d'ordres.

Définition 2.4 *si E est muni d'un préordre $\succeq$ alors l'extension multi-ensemble de cette relation est également un préordre appelé* ordre multi-ensemble *sur $\mathcal{M}(E)$, notée $\succeq_{mul}$.*

Proposition 2.5 *On a $M = \{s_1, \ldots, s_n\} \succeq_{mul} \{t_1, \ldots, t_m\} = N$ si et seulement si*

- *$M = N$ ou*
- *il existe i et j tels que $s_i \simeq t_j$ et $M - \{s_i\} \succeq_{mul} N - \{t_j\}$ ou*
- *il existe i, il existe $k \geq 0$ et une suite $j_1, \ldots, j_k$ tels que $1 \leq j_1 < j_2 < \cdots < j_k \leq m$, pour tout l, $s_i \succ t_{j_l}$ et $M - \{s_i\} \succeq_{mul} N - \{t_{j_1}, \ldots, t_{j_k}\}$.*

D'autre part, la relation d'équivalence associée à $\succeq_{mul}$ satisfait $\{s_1, \ldots, s_n\} \simeq_{mul} \{t_1, \ldots, t_m\}$ si et seulement si $n = m$ et il existe une permutation $\tau \in \mathfrak{S}_n$ telle que pour tout i, $s_i \simeq t_{\tau(i)}$.

2.1.2 Lemme de Higman

Nous rappelons maintenant un résultat sur les préordres qui est connu sous le nom de lemme de Higman [39]. Le lemme de Higman s'applique à la relation de divisibilité sur les mots. Nous nous intéressons dans la suite surtout au corollaire du lemme de Higman que nous donnons ensuite, qui concerne plutôt l'inclusion des multi-ensembles. Ce résultat sur les multi-ensembles sera utilisé au chapitre 5 et c'est pourquoi nous en rappelons une preuve complète. La démonstration que nous donnons ici est due à Nash-Williams [73].

Définition 2.6 *Un préordre $\preceq$ sur un ensemble $\mathcal{E}$ est un* pré-bel-ordre *si pour toute suite infinie $s_1, s_2, \ldots$ d'éléments de $\mathcal{E}$ il existe i et j tel que $i < j$ et $s_i \preceq s_j$.*

Définition 2.7 *Soit $\preceq$ un préordre sur $\mathcal{E}$ et soient $u = a_1 a_2 \cdots a_n$ et $v = b_1 b_2 \cdots b_m$ deux mots de $\mathcal{E}^*$. On dit que u* divise *v (ou encore que u est un sous-mot de v) si et seulement s'il existe une suite d'indices $1 \leq j_1 < \cdots < j_n \leq m$ telle que pour tout i, $a_i \preceq b_{j_i}$. On note alors $u \preceq_{div} v$.*

Notre but est de montrer que si $\preceq$ est un pré-bel-ordre alors $\preceq_{div}$ est un pré-bel-ordre. Nous montrons d'abord deux lemmes préliminaires.

Lemme 2.8 *Si $\preceq$ est un pré-bel-ordre sur $\mathcal{E}$ alors de toute suite $s_1, s_2, \ldots$ on peut extraire une sous-suite croissante :*

$$\exists i_1 < i_2 < \cdots, \quad s_{i_1} \preceq s_{i_2} \preceq \cdots$$

Démonstration. par l'absurde : on suppose qu'une certaine suite $s_1, s_2, \dots$ n'a pas de sous-suite croissante. Pour tout $j \geq 1$, on peut construire une séquence finie $s_{k_1}, s_{k_2}, \dots, s_{k_l}$ en prenant s_j pour premier terme et ensuite par récurrence en prenant à chaque fois pour élément suivant de s_{k_i} le premier terme qui est plus grand s'il existe. Cette séquence est toujours finie par hypothèse :

- $s_{k_1} = s_j$
- $\forall i,\ k_1 \leq i < k_l,\ s_{k_{i+1}}$ est le premier élément s_m de $s_{k_i+1}, s_{k_i+2}, \dots$ tel que $s_{k_i} \preceq s_m$.
- $\forall i > k_l,\ s_{k_l} \not\preceq s_i$

Appelons cette séquence la première séquence croissante maximale (abrégé en pscm) commençant à s_j.

Maintenant, nous pouvons définir par récurrence une suite

$$(s_{n_{1,1}}, \dots, s_{n_{1,k_1}}), (s_{n_{2,1}}, \dots, s_{n_{2,k_2}}), \dots$$

de séquences finies extraites de $s_1, s_2, \dots$ en prenant la pscm de s_1 et ensuite à chaque fois la pscm du successeur du dernier élément de la séquence précédente :

- $s_{n_{1,1}}, \dots, s_{n_{1,k_1}}$ est la pscm de s_1
- $\forall i > 1,\ s_{n_{i,1}}, \dots, s_{n_{i,k_i}}$ est la pscm de $s_{n_{i-1,k_{i-1}}+1}$

Considérons à présent la suite $s_{n_{1,k_1}}, s_{n_{2,k_2}}, \dots$: puisque $\preceq$ est un pré-bel-ordre, il existe $i < j$ tels que $s_{n_{i,k_i}} \preceq s_{n_{j,k_j}}$. Mais ceci est impossible car $s_{n_{i,k_i}}$ est le dernier élément d'une pscm. □

Lemme 2.9 *Soit $>$ un ordre strict noethérien sur un ensemble $\mathcal{E}$, et $\preceq$ un ordre sur $\mathcal{E}$ qui n'est pas un pré-bel-ordre. Alors l'ensemble des suites contre-exemples*

$$\mathcal{S} = \{s_1, s_2, \dots \mid \forall i < j,\ s_i \not\preceq s_j\}$$

possède au moins un élément minimal pour $>_{lex}$.

Il faut remarquer que $\mathcal{S}$ n'est pas vide puisque $\preceq$ n'est pas un pré-bel-ordre. Ce lemme n'est pas trivial car $>_{lex}$ n'est pas noethérien : si $s > t$ alors on a la suite décroissante

$$(s, s, s, \dots) >_{lex} (t, s, s, \dots) >_{lex} (t, t, s, \dots) >_{lex} \cdots$$

En fait, ce lemme est vrai parce que $\mathcal{S}$ est, au sens topologique, fermé.

Démonstration. Nous définissons simultanément par récurrence une suite $E_1, E_2, \dots$ de sous-ensembles non vides de $\mathcal{E}$, une suite $t_1, t_2, \dots$ d'éléments de $\mathcal{E}$ et une suite $F_0, F_1, F_2, \dots$ d'ensembles non vides de séquences d'éléments de $\mathcal{E}$:

- on prend $F_0 = \mathcal{S}$, qui, on l'a déjà remarqué, est non vide;

- ensuite si l'on suppose que, pour un $n \geq 0$, E_j et t_j ont été définis pour $1 \leq j < n$, et que F_j a été défini pour $0 \leq j < n$, posons

$$E_n = \{s_n \mid (s_1, s_2, \ldots) \in F_{n-1}\}$$

  E_n est non vide car F_{n-1} est non vide, donc il possède au moins un élément minimal pour $>$, nous choisissons pour t_n un tel élément minimal, et

$$F_n = \{(s_1, s_2, \ldots) \in F_{n-1} \mid s_n = t_n\}$$

  qui clairement est encore non vide.

On a toujours $F_n \subseteq F_{n-1}$ donc $F_n \subseteq F_0 = \mathcal{S}$. Maintenant on montre que la suite $t_1, t_2, \ldots$ est minimale dans $\mathcal{S}$. D'abord c'est une suite contre-exemple : si nous avons $i < j$ tel que $t_i \preceq t_j$ alors puisque F_j est non vide, prenons $(s_1, s_2, \ldots)$ dans F_j, donc dans $\mathcal{S}$, mais alors $s_i = t_i$ et $s_j = t_j$ donc $s_i \preceq s_j$, contradiction. Ensuite c'est une suite minimale dans $\mathcal{S}$: supposons qu'il y ait une suite $(u_1, u_2, \ldots) \in \mathcal{S}$ plus petite que $(t_1, t_2, \ldots)$ pour $>_{lex}$. Alors il existe N tel que $u_1 = t_1, \ldots, u_{N-1} = t_{N-1}, u_N < t_N$. Mais alors $(u_1, u_2, \ldots) \in F_{N-1}$ donc $u_N \in E_N$ et donc t_N n'est pas minimal dans E_N. $\square$

Remarque : comme E_n peut ne pas être dénombrable, nous avons besoin de l'axiome du choix pour choisir un t_n pour tout n. Néanmoins ce lemme peut être démontré par une méthode constructive [70].

Théorème 2.10 (Nash-Williams 1963) *Si $\preceq$ est un pré-bel-ordre sur un ensemble E alors $\preceq_{div}$ est un pré-bel-ordre sur E^*.*

Démonstration. supposons que $\preceq_{div}$ ne soit pas un pré-bel-ordre, et appliquons le lemme 2.9 à $\mathcal{E} = E^*$ et où $>$ est la comparaison des longueurs ($u > v$ si $|u| >_{\mathbb{N}} |v|$). Nous avons alors un contre-exemple minimal $u_1, u_2, \ldots$. Pour tout i, $u_i \neq \epsilon$ car si $u_i = \epsilon$ alors $u_i \preceq_{div} u_{i+1}$. Nous prenons alors la première lettre a_i de chacun des u_i pour tout i, et nous avons une suite $a_1, a_2, \ldots$ sur E. Comme $\preceq$ est un pré-bel-ordre sur E, il existe $i_1 < i_2 < \cdots$ tel que $a_{i_1} \preceq a_{i_2} \preceq \cdots$. Considérons alors la suite $u'_1, u'_2, \ldots$ suivante :

$$u_1, u_2, \ldots, u_{i_1-1}, u_{i_1} - a_{i_1}, u_{i_2} - a_{i_2}, \ldots$$

où $-$ est défini par $a_1 a_2 \cdots a_n - a_1 = a_2 \cdots a_n$. Cette suite est plus petite que $u_1, u_2, \ldots$ car le premier élément différent est u_{i_1} qui a été remplacé par $u_{i_1} - a_{i_1}$ donc la longueur est inférieure. Donc par minimalité cette suite n'est pas un contre-exemple : $\exists n < m,\ u'_n \preceq_{div} u'_m$. Nous avons alors trois cas suivant les positions respectives de i_1,n et m :

- si $n < m < i_1$ alors $u'_n = u_n$ et $u'_m = u_m$ donc $u_n \preceq_{div} u_m$, donc $u_1, u_2, \ldots$ n'est pas un contre-exemple;
- si $n < i_1 \leq m$ alors $u'_n = u_n$ et $u'_m = u_{i_k} - a_{i_k}$ pour un certain k mais alors $u_n \preceq_{div} u_{i_k} - a_{i_k} \preceq_{div} u_{i_k}$, donc $u_1, u_2, \ldots$ n'est pas un contre-exemple;
- si $i_1 \leq n < m$ alors $u'_n = u_{i_k} - a_{i_k}$ et $u'_m = u_{i_l} - a_{i_l}$ pour certains k et l, mais alors $u_{i_k} - a_{i_k} \preceq_{div} u_{i_l} - a_{i_l}$ et $s_{i_k} \preceq s_{i_l}$ donc $u_{i_k} \preceq_{div} u_{i_l}$, donc $u_1, u_2, \ldots$ n'est pas un contre-exemple. $\square$

Corollaire 2.11 *Si $\preceq$ est un pré-bel-ordre sur un ensemble E alors $\preceq_{mul}$ est un pré-bel-ordre sur $\mathcal{M}(E)$. En particulier, si F est un ensemble fini et $M_1, M_2, \ldots$ une suite de multi-ensembles d'éléments de F, alors il existe i et j tel que $M_i \subseteq M_j$.*

Démonstration. C'est une conséquence immédiate du résultat sur la divisibilité, puisque $\preceq_{div} \subseteq \preceq_{mul}$. Pour le cas particulier, il suffit de considérer l'ordre $=$ sur F, qui est un pré-bel-ordre si F est fini. □

2.1.3 Théorème de Kruskal

Le théorème de Kruskal [56, 57] va nous permettre de démontrer la bonne fondation des ordres que nous allons introduire.

Définition 2.12 *Soit $\mathcal{F}$ une signature ordonnée par un préordre $\preceq$. La relation de plongement $\trianglelefteq_{\preceq}$ (ou $\trianglelefteq$ s'il n'y a pas d'ambiguïté) sur $T(\mathcal{F})$ est définie par $f(s_1, \ldots, s_n) \trianglelefteq g(t_1, \ldots, t_m)$ si :*

- *il existe j tel que $f(t_1, \ldots, t_n) \trianglelefteq t_j$ ou*
- *$f \preceq g$ et $s_1 \cdots s_n \trianglelefteq_{div} t_1 \cdots, t_m$.*

Théorème 2.13 (Kruskal 1960) *Si $\preceq$ est un pré-bel-ordre, le plongement $\trianglelefteq_{\preceq}$ est un pré-bel-ordre.*

Démonstration. Raisonnons par l'absurde : considérons une suite contre-exemple $t_1, t_2, \ldots$ minimale pour la taille. Alors l'ensemble des sous-termes stricts des t_i est bien-ordonné pour le plongement : sinon il existerait une suite contre-exemple $s_1, s_2, \ldots$. Or on sait que s_1 est un sous-terme strict de t_l pour un certain l, et on peut supposer que pour tout i, s_i est un sous-terme de l'un des $t_l, t_{l+1}, \ldots$, quitte à supprimer de cette suite ceux qui sont sous-terme de $t_1, \ldots, t_{l-1}$, qui sont en nombre fini. Mais alors la suite $t_1, \ldots, t_{l-1}, s_1, s_2, \ldots$ est un contre-exemple plus petit que $t_1, t_2, \ldots$, une contradiction.

La signature étant bien-ordonnée, il existe une sous-suite de $t_1, t_2, \ldots$ telle que les symboles de tête forment une suite croissante pour $\preceq$. Les ensembles des sous-termes à profondeur 1 sont bien ordonnés, donc par le théorème 2.10, il existe deux termes t_i et t_j tels que leurs mots des sous-termes à profondeur 1 vérifient $u_1 \cdots u_n \trianglelefteq_{div} v_1 \ldots v_m$ et donc $t_i \trianglelefteq t_j$. □

Il faut faire attention que dans le cas d'un nombre infini de symboles, l'égalité elle-même n'est pas un pré-bel-ordre.

2.2 Ordres de réduction

Pour être utile à la réécriture, les ordres sur les algèbres de termes doivent vérifier un certain nombre de propriétés de compatibilités avec la structure de terme. Ce sont ces propriétés que nous rappelons dans cette section.

Définition 2.14 *Un préordre $\succeq$ sur $TV(\mathcal{F}, \mathcal{FV}, \mathcal{X})$ est un* ordre de réduction *si*

- *$\succeq$ est terme-compatible;*

- $\succ$ *est terme-compatible;*
- $\succeq$ *est bien-fondé;*

L'intérêt des ordres de réduction est justifié par la proposition suivante :

Proposition 2.15 *Un système de réécriture R termine si et seulement s'il existe un ordre de réduction $\succeq$ tel que pour toute règle $l \to r$ de R, $l \succ r$.*

Démonstration. Immédiate. Il faut remarquer que si R termine, la relation $\xrightarrow[R]{*}$ est un ordre de réduction. □

Proposition 2.16 *Les fonctionnelles lexicographique et multi-ensemble préservent la terme-compatibilité.*

Démonstration. Immédiate. Il faut quand même remarquer que l'on a effectivement besoin de la compatibilité à la fois du préordre et de l'ordre strict pour avoir la compatibilité des préordres et des ordres stricts étendus par lex ou mul. □

Corollaire 2.17 *Les fonctionnelles lexicographique et multi-ensemble transforment les ordres de réduction en ordres de réduction.*

Les ordres de réduction classiques font partie d'une sous-classe des ordres de réduction : les ordres de simplification.

Définition 2.18 *Un préordre $\succeq$ sur $TV(\mathcal{F}, \mathcal{FV}, \mathcal{X})$ est un* ordre de simplification *si*

- $\succeq$ *est terme-compatible;*
- $\succ$ *est terme-compatible;*
- $\succ$ *a la propriété sous-terme : si s est un sous-terme strict de t alors $s \succ t$.*

La propriété sous-terme ainsi définie sur les termes variadiques englobe à la fois la propriété sous-terme $f(\ldots, s, \ldots) \succ s$ et la propriété d'effacement $f(\ldots, s, \ldots) \succ f(\ldots, \ldots)$ de Dershowitz [28].

Théorème 2.19 (Dershowitz 82) *Sur une signature finie, tout ordre de simplification est bien-fondé (donc est un ordre de réduction).*

Démonstration. Par l'absurde. Supposons que l'on ait une suite strictement décroissante pour $\succeq$: $s_1 \succ s_2 \succ \cdots$. En instanciant toutes les variables par une constante arbitraire, on obtient une suite de termes clos (elle restera décroissante car l'ordre est stable par instanciation). Comme le plongement $\trianglelefteq_=$ est un pré-bel-ordre, on sait qu'il existe n et m, $n < m$, tels que $s_n \trianglelefteq s_m$, mais ceci contredit $s_n \succ s_m$ puisque $\succeq$ contient le plongement. □

De manière à tester la terminaison de la réécriture modulo une théorie équationnelle E, on utilise un ordre de réduction qui soit en plus compatible avec E. En fait, une condition plus faible, dite de *commutation* [46], est suffisante, mais nous n'en aurons pas besoin.

Définition 2.20 $\succeq$ *est* compatible avec une théorie E *(ou est* E-compatible*) si pour tout* s*,* s'*,* t *et* t' *tels que* $s =_E s'$ *et* $t =_E t'$*,* $s \succeq t \Longrightarrow s' \succeq t'$ *et* $s \succ t \Longrightarrow s' \succ t'$*.*

Lemme 2.21 *Soit* $\succeq$ *un préordre et* $\simeq$ *sa relation d'équivalence associée. Si* $=_E \subseteq \simeq$ *alors* $\succeq$ *est compatible avec* E*.*

Démonstration. Évidente. □

Lemme 2.22 *Soit* R *un ensemble de règles. S'il existe un ordre de simplification* $\succeq$ *compatible avec* E *et tel que pour toute règle* $l \to r$ *de* R *on ait* $l \succ r$*, alors* $\xrightarrow[R/E]{}$ *termine.*

Démonstration. Pour toute réduction $s \xrightarrow[l\to r/E]{} t$, on a $s =_E s'$ et $t =_E t'$ avec $s' \xrightarrow[l\to r]{} t'$ donc $s' \succ t'$ et par compatibilité $s \succ t$. □

Malheureusement, il n'existe en général pas, pour une théorie E arbitraire, d'ordre de simplification compatible avec E. Par exemple, il n'existe pas d'ordre compatible avec ACI exempté l'égalité ! En effet, si $+$ est un opérateur ACI et à partir du moment où il y a deux termes s et t tels que $s \succ t$, alors on a :

$$s =_{ACI} s + s \succ s + t =_{ACI} s + s + t \succ \cdots$$

ce qui contredit la bonne fondation.

Par contre il existe des ordres compatibles avec l'associativité et la commutativité, ce qui est évidemment fondamental car nous allons utiliser essentiellement de la réécriture modulo AC. La définition de tels ordres compatible avec AC fait l'objet des deux dernières sections de ce chapitre, la dernière étant consacrée aux ordres AC totaux.

2.3 Exemples fondamentaux d'ordres de simplification

Nous commençons par rappeler les définitions et propriétés des ordres de simplification classiques.

2.3.1 Ordres engendrés par interprétations polynomiales

Les interprétations polynomiales ont été historiquement introduites par Lankford [59,60]. À l'origine, elles étaient à valeurs dans $\mathbb{N}$. Nous aurons besoin dans la suite d'interpréter dans des ensembles plus généraux, à savoir dans des polynômes à coefficients dans $\mathbb{N}$.

Attention : pour les ordres engendrés par des interprétations polynomiales, on se restreint aux termes non variadiques.

Définition 2.23 *On utilisera comme* domaine d'interprétation *l'ensemble de polynômes* $D = \mathbb{N}[X_i, i \in I]$ *où* $(X_i)_{i\in I}$ *est une famille finie d'indéterminées, que l'on munit d'un ordre* $>$ *bien-fondé compatible avec la substitution de polynômes. Nous nous bornerons à donner deux définitions possibles, où dans les deux l'on compare deux polynômes en comparant les multi-ensembles de leurs monômes par l'extension multi-ensemble de deux variantes d'ordres sur les monômes. On se donne d'abord une précédence sur les indéterminées,* $X_1 > \cdots > X_k$*, et on compare les monômes par deux techniques :*

1. *Deux monômes sont comparés par l'extension lexicographique de leurs exposants, et en cas d'égalité par leurs coefficients :*

$$\begin{array}{ccc} aX_1^{n_1}\dots X_k^{n_k} & > & bX_1^{m_1}\dots X_k^{m_k} \iff \\ (n_1,\dots,n_k,a) & (>_{\mathbb{N}})_{lex} & (m_1,\dots,m_k,b) \end{array}$$

2. *On compare les degrés totaux d'abord :*

$$\begin{array}{ccc} aX_1^{n_1}\dots X_k^{n_k} & > & bX_1^{m_1}\dots X_k^{m_k} \iff \\ (\sum_{i=1}^{k} n_i, n_1,\dots,n_{k-1},a) & (>_{\mathbb{N}})_{lex} & (\sum_{i=1}^{k} m_i, m_1\dots,m_{k-1},b) \end{array}$$

Ces ordres sont bien-fondés car construits par extension lexicographique puis multi-ensemble d'un ordre bien-fondé, et sont totaux (si la précédence est totale).

De manière à définir des ordres sur les termes avec variables, nous allons interpréter les termes comme des fonctions polynomiales sur le domaine D, dont les arguments sont les variables. Notons $D^{\mathcal{X}}$ l'ensemble des fonctions polynomiales de $\mathcal{X}$ dans D. Pour que l'ordre que l'on va définir sur les termes soit stable par instanciation, nous avons besoin de mettre sur cet ensemble un ordre stable par substitution de polynômes. Il y a plusieurs définitions possibles :

1. Ordre d'évaluation : $P > Q$ si et seulement si pour toute substitution $\sigma : \mathcal{X} \to D$ on a $P\sigma > Q\sigma$. Cet ordre est par définition stable par substitution. Le problème est qu'il n'est pas évident en pratique de tester $P > Q$. Un certain nombre de critères existent, dans le cas où le domaine D est $\mathbb{N}$ [13].

2. Ordre produit : $P > Q$ si pour chaque monôme de Q il existe un monôme de P (différent à chaque fois) qui soit supérieur pour l'ordre produit des exposants et du coefficient (voir exemple). Cet ordre n'est stable par substitution qu'à la condition d'interpréter dans $D - \{0,1\}$ (pour avoir $x > 1$ pour toute variable x), ce qu'on supposera.

Ces deux ordres ne sont bien sûr pas totaux car une variable ne peut pas se comparer à une autre variable. Néanmoins, il faut remarquer qu'ils sont totaux sur les termes clos (si l'ordre sur D est total bien sûr).

Définition 2.24 *Une* interprétation polynomiale *est la donnée pour chaque $f \in \mathcal{F}$ d'arité n d'une fonction polynomiale ϕ_f à n variables : $D^n \to D$, qu'on supposera de degré au moins 1 en chacune des variables. Une interprétation polynomiale engendre une application I de $T(\mathcal{F},\mathcal{X})$ dans $D^{\mathcal{X}}$ définie de la façon suivante :*

$$\begin{cases} I(f(s_1,\dots,s_n)) = \phi_f(I(s_1),\dots,I(s_n)) \\ I(x) = x \end{cases}$$

L'ordre polynomial *engendré par I est la relation $\succeq_I$ sur $T(\mathcal{F},\mathcal{X})$ définie par $s \succeq_I t$ si et seulement si $I(s) \leq I(t)$.*

Lemme 2.25 *Pour tout I, $\succeq_I$ est un ordre de simplification.*

Démonstration. Vérifions chacune des propriétés requises :

- stabilité par contexte : si $s \succ_I t$ alors pour tout f d'arité n et tous termes $u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_n$ on a

$$I(f(u_1, \ldots, u_{i-1}, s, u_{i+1}, \ldots, u_n)) = \phi_f(I(u_1), \ldots, I(u_{i-1}), I(s), I(u_{i+1}), \ldots, I(u_n))$$

  et de même pour t. Comme ϕ_f est de degré au moins 1 en chacune des variables, c'est une fonction strictement croissante en chacune des variables, donc on a bien

$$f(u_1, \ldots, u_{i-1}, s, u_{i+1}, \ldots, u_n) \succ_I f(u_1, \ldots, u_{i-1}, t, u_{i+1}, \ldots, u_n).$$

  On a le même résultat pour le préordre $\succeq_I$.

- stabilité par instanciation : si $s \succ_I t$ et σ est une substitution arbitraire, alors $I(s\sigma) = I(s)\sigma$ en considérant σ comme une substitution de polynômes, et donc $I(s\sigma) \succ_I I(t\sigma)$. On a le même résultat pour le préordre $\succeq_I$.

- propriété sous-terme : évidente. □

Exemple. Montrons que le système de règles suivant (associativité et distributivité) termine [60] :

$$\begin{cases} (x+y)+z \to x+(y+z) \\ x*(y+z) \to (x*y)+(x*z) \\ (x+y)*z \to (x*z)+(y*z) \end{cases}$$

On utilise l'interprétation suivante dans $D = \mathbb{N}$

$$\begin{cases} \phi_+(X,Y) = 2X+Y+1 \\ \phi_*(X,Y) = XY \end{cases}$$

Pour la règle d'associativité on a

$$\begin{cases} I((x+y)+z) = 4x+2y+z+3 \\ I(x+(y+z)) = 2x+2y+z+2 \end{cases}$$

Le premier polynôme est plus grand que le deuxième pour l'ordre produit ($4x > 2x$ et $3 > 2$). Pour les autres règles, par exemple la deuxième :

$$\begin{cases} I(x*(y+z)) = 2xy+xz+x \\ I((x*y)+(x*z)) = 2xy+xz+1 \end{cases}$$

et encore une fois le premier polynôme est inférieur au second pour l'ordre produit ($x > 1$)

2.3.2 Ordre récursif sur les chemins

La notion d'ordre récursif sur les chemins a été introduite simultanément par David Plaisted [79] et Nachum Dershowitz [26, 31] en 1978, ces versions utilisant l'ordre multi-ensemble. Ensuite l'ordre récursif sur les chemins dit *lexicographique* a été introduit par Kamin et Lévy en 1980 [47], et enfin Dershowitz a donné une définition à la fois plus générale et plus agréable, englobant les deux définitions précédentes, en 1982 [27]. Nous donnons ici la définition la plus générale de l'ordre récursif sur les chemins : sur une signature pouvant être infinie, et avec statuts lexicographique ou multi-ensemble.

Définition 2.26 *Une* précédence *est un préordre sur $\mathcal{F} \cup \mathcal{FV}$. Un* statut *est un élément de l'ensemble $\{MUL, LEX\}$. MUL s'appelle le statut multi-ensemble, LEX s'appelle le statut lexicographique. Une application* st *de $\mathcal{F} \cup \mathcal{FV}$ dans $\{MUL, LEX\}$ est une fonction de statut admissible pour une précédence $\succeq$ si :*

- *pour tout f de $\mathcal{FV}$, $\mathrm{st}(f) = MUL$;*
- *pour tous f et g de $\mathcal{F} \cup \mathcal{FV}$, $f \simeq g \Longrightarrow \mathrm{st}(f) = \mathrm{st}(g)$ et si de plus $\mathrm{st}(f) = LEX$, alors $\mathrm{ar}(f) = \mathrm{ar}(g)$.*

Remarque : on peut donner une définition plus générale encore en associant à chaque opérateur de statut lexicographique d'arité n une permutation de $\mathfrak{S}_n$. En particulier, les opérateurs auxquels est associé la permutation identité sont dits à statut lexicographique gauche-droite, et ceux auxquels est associé la permutation « miroir » $1 \to n, 2 \to n-1, \ldots, n \to 1$ sont dits à statut lexicographique droite-gauche.

Définition 2.27 *Soit $\succeq$ une précédence sur $\mathcal{F}$ et* st *une fonction de statut admissible pour $\succeq$. On appelle* ordre récursif sur les chemins *(en anglais* recursive path ordering*) la relation $\succeq_{rpo}$ sur $TV(\mathcal{F}, \mathcal{FV}, \mathcal{X})$ définie récursivement par $s \succeq_{rpo} t$ si et seulement si*

- *$s = x \in \mathcal{X}$ et $t = x$ ou*
- *$s = f(s_1, \ldots, s_n)$ pour un $f \in \mathcal{F} \cup \mathcal{FV}$ et*
  - *$\exists i,\ s_i \succeq_{rpo} t$ ou*
  - *$t = g(t_1, \ldots, t_m)$ et*
    - *$f \succ g$ et $\forall j,\ s \succ_{rpo} t_j$ ou*
    - *$f \simeq g$ (donc $\mathrm{st}(f) = \mathrm{st}(g)$) et*
      - *$\mathrm{st}(f) = MUL$ et $\{s_1, \ldots, s_n\}(\succeq_{rpo})_{mul}\{t_1, \ldots, t_m\}$ ou*
      - *$\mathrm{st}(f) = LEX$ (donc $n = m$), $\forall j,\ s \succ_{rpo} t_j$, et $(s_1, \ldots, s_n)(\succeq_{rpo})_{lex}(t_1, \ldots, t_n)$*

Remarque : dans le cas général où est associé à chaque opérateur de statut lexicographique f une permutation π_f, on remplace la dernière ligne par

$$(s_{\pi_f(1)}, \ldots, s_{\pi_f(n)})(\succeq_{rpo})_{lex}(t_{\pi_g(1)}, \ldots, t_{\pi_g(n)}).$$

Proposition 2.28 (Dershowitz 1982) *La relation $\succeq_{rpo}$ est un préordre sur TV, monotone et stable par instanciation.*

Si la précédence est bien-fondée (en particulier si $\mathcal{F} \cup \mathcal{FV}$ est fini), $\succeq_{rpo}$ est bien-fondé, c'est donc un préordre de réduction.

Si la précédence est totale, et si tous les opérateurs ont le statut lexicographique, alors $\succeq_{rpo}$ est un ordre total sur l'ensemble des termes clos $TV(\mathcal{F}, \mathcal{FV})$.

2.4 Ordres AC-compatibles

Les ordres de réduction compatible avec AC jouent un rôle primordial, dans la mesure où de tels ordres sont indispensables pour la complétion associative-commutative. La définition de tels ordres a suscité beaucoup de travaux dont nous rappelons ici les résultats.

2.4.1 Ordres AC engendrés par interprétations polynomiales

Une première classe d'ordres AC-compatibles peut être définie par des interprétations polynomiales utilisant des polynômes de la forme appropriée. Les interprétations polynomiales n'étant définies que sur les termes non variadiques, on ne peut pas aplatir les termes par rapport aux symboles AC pour les comparer. On conserve donc les termes non aplatis, et l'on a une condition sur les polynômes associés aux symboles AC qui permette de rendre l'ordre polynomial associé compatible avec AC.

Définition 2.29 *Soit $+$ un symbole AC. On dit que ϕ_+ est un polynôme associatif-commutatif si*

$$\begin{cases} \phi_+(\phi_+(X,Y),Z) = \phi_+(X,\phi_+(Y,Z)) \\ \phi_+(X,Y) = \phi_+(Y,X) \end{cases}$$

Proposition 2.30 *Si ϕ_f est un polynôme AC pour tous les symboles AC f, alors l'ordre polynomial engendré est AC-compatible.*

Démonstration. Évidente. □

La condition nécessaire et suffisante suivante est due à Ben Cherifa et Lescanne [13].

Proposition 2.31 (Ben Cherifa et Lescanne 87) *Le polynôme ϕ_+ est associatif-commutatif si et seulement si c'est un polynôme symétrique du premier degré en chacune des variables $aXY + b(X+Y) + c$ tel que $b^2 = b + ac$.*

Démonstration. Supposons $P(X,Y) = \phi_+(X,Y)$ compatible avec AC. La condition de commutativité impose qu'il soit symétrique en X et Y, donc $P(X,Y) = Q(XY, X+Y)$. D'autre part, la condition d'associativité impose la relation sur les degrés $d^\circ_X(P(P(X,Y),Z)) = d^\circ_X(P(X,P(Y,Z)))$ donc $d^\circ_X(P)^2 = d^\circ_X(P)$ donc P est au plus de degré 1 en X; et de même en Y puisqu'il est symétrique. P est donc de la forme $P(X,Y) = aXY + b(X+Y) + c$. Écrivons à nouveau la condition d'associativité :

$$\begin{aligned} a\,[aXY + b(X+Y) + c]\,Z + b\,[aXY + b(X+Y) + c + Z] + c \quad &= \\ aX\,[aYZ + b(Y+Z) + c] + b\,[X + aYZ + b(Y+Z) + c] + c & \end{aligned}$$

d'où après développement :

$$\begin{aligned} a^2XYZ + abXZ + abYZ + baXY + b^2X + b^2Y + (ac+b)Z + bc + c \quad &= \\ a^2XYZ + abXY + abXZ + baYZ + (ac+b)X + b^2Y + b^2Z + bc + c & \end{aligned}$$

donc en identifiant les coefficients $b^2 = b + ac$.

La réciproque est évidente. □

Exemple. Montrons la terminaison du système de réécriture classique pour l'addition des entiers naturels :

$$\begin{cases} AC(+) \\ x + 0 \to x \\ x + s(y) \to s(x+y) \end{cases}.$$

On prend les interprétations :

$$\begin{cases} \phi_0 = 2 \\ \phi_s(X) = X + 1 \\ \phi_+(X, Y) = XY \end{cases} .$$

Le polynôme ϕ_+ vérifie bien la condition $b^2 = b + ac$ et on a :

$$\begin{cases} I(x + 0) = 2x > x = I(x) \\ I(x + s(y)) = xy + y > xy + 1 = I(s(x + y)) \end{cases} .$$

donc le système termine modulo AC.

Exemple. Système canonique des anneaux commutatifs.

$$CR \begin{cases} AC(*) \\ AC(+) \\ x + 0 \to x \\ x + (-x) \to 0 \\ -0 \to 0 \\ -(-x) \to x \\ -(x + y) \to (-x) + (-y) \\ x * 1 \to x \\ x * (y + z) \to (x * y) + (x * z) \\ x * 0 \to 0 \\ x * (-y) \to -(x * y) \end{cases} .$$

On utilise l'interprétation

$$\begin{cases} \phi_0 = \phi_1 = 2 \\ \phi_+(X, Y) = X + Y + 4 \\ \phi_*(X, Y) = XY + 2(X + Y) + 2 \\ \phi_-(X) = 2X + 3 \end{cases} .$$

Comme on peut le voir sur ce dernier exemple, trouver une interprétation polynomiale adaptée demande une certaine expertise, cela ne peut pas se trouver (dans l'état actuel des connaissances) de manière automatique, ou même semi-automatique. C'est pourquoi on préfère utiliser des ordres basés sur une précédence sur les opérateurs.

2.4.2 Ordres AC basés sur l'ordre récursif sur les chemins

Une idée naturelle serait d'utiliser l'ordre récursif sur les chemins sur les termes aplatis avec un statut multi-ensemble pour les opérateurs AC. Malheureusement, un tel ordre n'est pas monotone à cause de l'aplatissement. Un exemple classique est le suivant :

Exemple. supposons que l'on ait deux opérateurs AC $+$ et $*$ avec $* > +$. On a alors pour une constante a arbitraire $a * a \succ_{rpo} a + a$, donc par monotonie on devrait avoir $a * (a * a) \succ_{rpo} a * (a + a)$ mais pour comparer ces deux termes par le *rpo*, on est amené à comparer les multi-ensembles $\{a, a, a\}$ et $\{a, a + a\}$ et c'est le second qui est le plus grand ! on a donc $a * a * a \prec_{rpo} a * (a + a)$.

En fait, on a le résultat suivant :

Proposition 2.32 *Si tous les opérateurs AC sont minimaux dans la précédence, alors $\succeq_{rpo}$ est compatible avec AC.*

Si l'on veut alors avoir un ordre total, puisque les opérateurs AC doivent être minimaux, cela impose qu'il n'y ait qu'un seul opérateur AC ! Cela donne donc un résultat très restrictif, mais néanmoins utile car il nous servira pour le cas des groupes abéliens :

Proposition 2.33 *Si la signature ne contient qu'un seul opérateur AC $+$, que la précédence est totale, que $+$ est l'élément minimal de la précédence, que les opérateurs ont tous un statut lexicographique sauf $+$ qui a le statut multi-ensemble, alors $\succeq_{rpo}$ est un ordre de simplification compatible avec AC sur TV, total sur les termes clos.*

Exemple. Montrons la terminaison du système classique pour les groupes abéliens :

$$\left\{\begin{array}{l} \mathrm{AC}(+) \\ x+0 \rightarrow x \\ x+(-x) \rightarrow 0 \\ x+(-x)+y \rightarrow y \\ -(x+y) \rightarrow (-x)+(-y) \\ -0 \rightarrow 0 - (-x) \rightarrow x \end{array}\right. .$$

On vérifie aisément que les règles décroissent pour le *rpo* avec la précédence $- > 0 > +$.

2.4.3 Ordre associatif sur les chemins

L'ordre associatif sur les chemins (en abrégé *apo* comme *associative path ordering*) est un ordre de simplification AC supportant plusieurs symboles AC. La version originale de cet ordre est due à Bachmair et Plaisted [10], version améliorée ensuite par Bachmair et Dershowitz [4, 5] et une version encore plus élaborée à été décrite par Delor et Puel [24, 25]. Nous donnons ici cette dernière version car elle nous est nécessaire pour traiter le cas des anneaux commutatifs.

Définition 2.34 *Pour des symboles arbitraires f et g non constants, notons $D_i(f,g)$ la $i^{\text{ème}}$règle de distributivité de f par rapport à g :*

$$\begin{array}{l} f(x_1,\ldots,x_{i-1},g(y_1,\ldots,y_k),x_{i+1},\ldots,x_n) \rightarrow \\ \quad g(f(x_1,\ldots,x_{i-1},y_1,x_{i+1},\ldots,x_n),\ldots,f(x_1,\ldots,x_{i-1},y_k,x_{i+1},\ldots,x_n)) \end{array}$$

On note $D(f,g)$ le système de règles $(D_i(f,g))_{1\le i\le \mathrm{ar}(f)}$.

Supposons donnée une précédence $>$ et notons $\mathcal{D}_>$ l'ensemble des règles distributives $D(f,g)$ pour tous les symboles non constants f et g tels que $f > g$ et f est AC ou bien inférieur à un symbole AC. Afin de donner la définition de l'ordre *apo*, il nous faut assurer la propriété de forme normale unique pour ce système. Ceci n'est vrai que si la précédence vérifie certaines conditions. Il y a en fait plusieurs conditions de précédence admissibles, nous nous limitons ici à décrire celle qui nous intéresse.

Lemme 2.35 (Delor et Puel 1992) *Si la précédence est de la forme $\mathcal{F} > * > u_k > \cdots > u_1 > + > a$ où $*$ et $+$ sont AC, les u_i sont unaires et a une constante, alors $\mathcal{D}_>$ est convergent modulo AC.*

Démonstration. Nous donnons la preuve car elle introduit un ordre qui va nous resservir plus tard, l'ordre qui permet de prouver la terminaison de $\mathcal{D}_>$: il s'agit simplement de montrer la décroissance des règles pour l'ordre polynomial $\succ_D$ engendré par

$$\begin{cases} \phi_*(X,Y) = XY \\ \phi_+(X,Y) = X+Y+2 \\ \phi_{u_i}(X) = (i+1)X+1 \end{cases}$$

La confluence se montre par la confluence locale, c'est-à-dire en calculant les paires critiques et en montrant qu'elles sont joignables. $\square$

Nous supposons dorénavant que la précédence vérifie la condition du lemme précédent, et nous notons $s\downarrow_>$ la forme normale d'un terme s pour le système $\mathcal{D}_>$. Pour obtenir la définition de l'ordre *apo*, il suffit maintenant de comparer par le *rpo* les formes normales aplaties :

Définition 2.36 *On définit la relation $\succeq_{apo}$ par $s \succeq_{apo} t$ si et seulement si $\overline{s\downarrow_>} \succeq_{rpo} \overline{t\downarrow_>}$.*

Théorème 2.37 (Delor et Puel 1992) *La relation $\succeq_{apo}$ est un ordre de simplification AC-compatible [25].*

L'ordre *apo* ainsi défini a une congruence associée qui est exactement l'égalité modulo ACD. Par conséquent, un tel ordre ne permet pas de prouver la terminaison par exemple du système canonique standard pour la théorie des anneaux (voir exemple), puisque celui-ci contient la règle de distributivité. Mais on peut résoudre ce problème en faisant une composition lexicographique avec un autre ordre : lorsque deux termes sont équivalents pour l'ordre associatif sur les chemins, alors on les compare par l'ordre polynomial $\succ_D$ de la preuve du lemme précédent, celui qui fait décroître les règles de distributivité.

On peut remarquer qu'il n'est pas possible d'accepter dans la précédence plus d'une constante inférieure à un symbole AC. En effet, si l'on a une précédence $+ > a > b$, alors on a $b{+}b \succ_{apo} a$ mais en appliquant le contexte $b{+}[.]$, l'inégalité change de sens car $b{+}b{+}b \prec_{rpo} b{+}a$ (car $\{b,b,b\} \prec_{mul} \{b,a\}$).

Remarque: pour traiter ce cas avec plusieurs constantes, on peut probablement s'en sortir avec une règle de « distributivité » pour les constantes : $f(\ldots,a,\ldots) \to f(\ldots,\perp,\ldots)$ où $\perp$ est la plus petite constante, comme le font Nieuwenhuis et Rubio (voir plus loin).

Exemple. Terminaison du système canonique des anneaux :

$$\begin{cases} x+0 \to x \\ x+(-x) \to 0 \\ -0 \to 0 \\ -(-x) \to x \\ -(x+y) \to (-x)+(-y) \\ x*1 \to x \\ x*(y+z) \to (x*y)+(x*z) \\ x*0 \to 0 \\ x*(-y) \to -(x*y) \end{cases}$$

On prend comme précédence $* > - > + > 0$, qui satisfait bien la condition.

2.5 Ordres totaux modulo AC

Nous aurons besoin pour les complétions closes d'avoir un ordre AC total sur les termes clos, afin d'éviter les échecs de la complétion. La définition d'un ordre AC total est également utile dans la complétion ordonnée [3] et dans les techniques de preuves par réfutation (résolution et paramodulation orientée modulo AC) [2, 82]. Une autre application est la décidabilité de l'unification modulo une théorie AC-close [72]

2.5.1 Totalité sur les classes modulo AC

Définition 2.38 *Un préordre $\preceq$ sur T est un* ordre AC total *si c'est un préordre total dont la congruence associée est exactement l'égalité modulo AC. Autrement dit : $s =_{\mathrm{AC}} t$ implique $s \simeq t$ et $s \neq_{\mathrm{AC}} t$ implique $s \prec t$ ou $t \prec s$.*

L'existence d'un tel ordre en présence de plusieurs symboles AC est resté un problème ouvert jusqu'en 1991, quand Narendran et Rusinowitch [71] ont donné une première solution, qui fonctionne grâce à une remarque anodine mais qui s'avère fondamentale, la préservation du symbole de tête d'un terme par interprétation. Puis Nieuwenhuis et Rubio en 1993 [76] ont donné une autre solution par une méthode plus agréable puisque basée sur une précédence plutôt que sur des interprétations polynomiales, mais qui utilise aussi la remarque fondamentale précédente. Nous présentons maintenant notre propre méthode, qui nous permet de plonger les deux définitions dans un même cadre générique.

2.5.2 Ordre AC engendré par une interprétation AC

Définition 2.39 *Une* interprétation AC *est une application $I : T(\mathcal{F}, X) \to D$ où D est muni d'un ordre bien-fondé $\succ$, qui vérifie les conditions suivantes :*

(1) compatibilité avec AC : $s =_{\mathrm{AC}} t \Longrightarrow I(s) = I(t)$;

(2) préserve le symbole de tête : $I(s) = I(t) \Longrightarrow \mathcal{H}\mathrm{ead}(s) = \mathcal{H}\mathrm{ead}(t)$;

(3) terme-compatibilité : si $I(s) \succ I(t)$ (resp. $\succeq$) alors pour tout symbole f, $I(f(\ldots,s,\ldots)) \succ I(f(\ldots,t,\ldots))$ (resp. $\succeq$), et pour toute substitution σ, $I(s\sigma) \succ I(t\sigma)$ (resp. $\succeq$).

(4) contient la relation sous-terme au sens large : pour tous s et f, $I(f(\ldots,s,\ldots)) \succeq I(s)$.

À partir d'une telle interprétation I, on peut définir un ordre AC de la façon suivante.

Définition 2.40 *L'ordre AC engendré par I est la relation $\preceq_I$ sur T définie par $s \preceq_I t$ si et seulement si*

- *$I(s) \prec I(t)$ ou*
- *$I(s) = I(t)$ et si $f = \mathcal{H}\mathrm{ead}(s)$ ($= \mathcal{H}\mathrm{ead}(t)$ par la condition (2)) alors*
  - *$f \notin \mathcal{F}_{\mathrm{AC}}$ et*
$$(s_1,\ldots,s_n)(\preceq_I)_{lex}(t_1,\ldots,t_n)$$
    où $s = f(s_1,\ldots,s_n)$ et $t = f(t_1,\ldots,t_n)$

– $f \in \mathcal{F}_{\mathrm{AC}}$ *et*

$$\{s'_1, \ldots, s'_n\}(\preceq_I)_{mul}\{t'_1, \ldots, t'_m\}$$

où $\overline{s} = f(s'_1, \ldots, s'_n)$ *et* $\overline{t} = f(t'_1, \ldots, t'_m)$.

Théorème 2.41 *Si I est une interprétation AC, la relation $\preceq_I$ est un préordre*

- *terme-compatible, ainsi que $\prec_I$;*
- *compatible avec AC;*
- *total sur les classes de congruences modulo AC;*
- *qui contient la relation sous-terme.*

C'est donc un ordre de simplification AC total.

Démonstration. Nous montrons successivement chacune des propriétés requises.

- Transitivité : on montre la propriété

$$\text{pour tout } s, t \text{ et } u \text{ tels que } \begin{cases} |s| \leq k \\ |t| \leq k \\ |u| \leq k \end{cases}, \begin{cases} s \preceq_I t \\ t \preceq_I u \end{cases} \Longrightarrow s \preceq_I u$$

  par récurrence sur k. Le cas $k = 1$ est trivial. Supposons la propriété prouvée pour tout $j < k$ et prouvons-la pour k : le seul cas non trivial est celui où $I(s) = I(t)$ et $I(t) = I(u)$. Mais par la condition (2) nous savons que s, t et u ont le même symbole de tête f, donc $s = f(s_1, \ldots, s_n)$, $t = f(t_1, \ldots, t_n)$ et $u = f(u_1, \ldots, u_n)$. Si $f \notin \mathcal{F}_{\mathrm{AC}}$ alors $(s_1, \ldots, s_n)(\preceq_I)_{lex}(t_1, \ldots, t_n)$ et $(t_1, \ldots, t_n)(\preceq_I)_{lex}(u_1, \ldots, u_n)$, donc par récurrence $(s_1, \ldots, s_n)(\preceq_I)_{lex}(u_1, \ldots, u_n)$. Le cas $f \in \mathcal{F}_{\mathrm{AC}}$ est similaire, avec la fonctionnelle multi-ensemble au lieu de lexicographique.

- Stabilité par mise sous contexte : supposons $s \preceq_I t$ et soit $f(\ldots, ., \ldots)$ un contexte quelconque. Si $I(s) \prec I(t)$ alors on conclut par (3). Si $I(s) = I(t)$, alors par la propriété (3) $I(f(\ldots, s, \ldots)) \succeq I(f(\ldots, t, \ldots))$, s'il n'y a pas égalité c'est clair, sinon on a deux cas :
  - si $f \notin \mathcal{F}_{\mathrm{AC}}$, clairement $(\ldots, s, \ldots)(\preceq_I)_{lex}(\ldots, t, \ldots)$;
  - si $f \in \mathcal{F}_{\mathrm{AC}}$, il faut regarder le symbole de tête g de s et t (c'est le même par (2)) : si $f \neq g$ alors on conclut comme dans le premier cas, et si $f = g$ alors $\{s_1, \ldots, s_n\}(\preceq_I)_{mul}\{t_1 \ldots, t_m\}$ et on conclut facilement car $\mathrm{flat}(f(\ldots, s, \ldots)) = f(\ldots, s_1, \ldots, s_n, \ldots)$ et de même pour t. Le raisonnement est identique pour la monotonie de l'ordre strict.

- Stabilité par instanciation : par récurrence sur la taille de s et t : si $s \prec_I t$ alors si $I(t) \succ I(s)$ on conclut par (3), et si $I(s) = I(t)$ on applique l'hypothèse de récurrence aux sous-termes de s et t.

- Compatibilité avec AC : évidente par (1).

- Totalité : il faut montrer que pour tout s et t, soit $s \prec_I t$, soit $t \prec_I s$, soit $s =_{\mathrm{AC}} t$. Or, si l'on n'est pas dans l'un des deux premiers cas, alors nécessairement $I(s) = I(t)$, et on conclut par récurrence évidente.

- Sous-terme : évidente par la propriété sous-terme de I.

2.5.3 Exemples d'interprétations AC

Nous donnons maintenant des exemples d'interprétations AC. Les premières sont des interprétations polynomiales : la première interprète les termes dans l'ensemble $\mathbb{N}$ des entiers naturels muni de l'ordre habituel. La deuxième, la première introduite historiquement par Narendran et Rusinowitch [71] en 1991, interprète les termes dans un anneau de polynômes à plusieurs indéterminées à coefficients dans $\mathbb{Z}$. La dernière a été découverte récemment par Nieuwenhuis et Rubio [76], en 93, et interprète dans l'algèbre des termes elle-même, par calcul d'une forme normale.

Interprétations dans les entiers naturels

On prend $D = \mathbb{N}^X$. Supposons que l'ensemble des opérateurs est indexé $\mathcal{F} = \{f_1, \ldots, f_N\}$. On définit I de la façon suivante :

$$\begin{cases} I(f_i(s_1, \ldots, s_n)) = a_i I(s_1) \ldots I(s_n) + b_i(I(s_1) + \ldots + I(s_n)) + c_i \text{ où} \\ a_i = N(Ni - 1) \\ b_i = Ni \\ c_i = i \\ I(x) = x \end{cases}.$$

Vérifions les quatre propriétés :

(1) il suffit que $b_i^2 = b_i + a_i c_i$, ce qui est trivial;

(2) $I(f_i(s_1, \ldots, s_n)) \equiv i \pmod N$ donc $\mathcal{H}\text{ead}(s) = f_{I(s) \bmod N}$;

(3) vrai par le lemme 2.25.

(4) trivial.

En fait, on peut généraliser cette interprétation par

$$\begin{cases} a_i = N(N(i + d_i) - 1) \\ b_i = N(i + d_i) \\ c_i = i + d_i \end{cases}$$

où chaque d_i peut être choisi arbitrairement.

Cette interprétation suppose fini et connu le nombre d'opérateurs. En pratique, cela pose un problème car on peut avoir envie d'ajouter des nouveaux opérateurs, et même avoir besoin d'une infinité d'opérateurs. Nous donnons à présent une interprétation qui accepte une infinité d'opérateurs $\mathcal{F} = \{f_1, f_2, \ldots\}$. Définissons I comme suit :

$$\begin{cases} I(f_i(s_1, \ldots, s_n)) = a_i I(s_1) \ldots I(s_n) + b_i(I(s_1) + \ldots + I(s_n)) + c_i \text{ où} \\ a_i = 2^{i+1}(2^{2i+1} - 1) \\ b_i = 2^{2i+1} \\ c_i = 2^i \\ I(x) = x \end{cases}.$$

On vérifie les propriétés requises de la même manière que précédemment. La propriété (2) est vraie car $v_2(I(f_i(\ldots))) = i$ donc $\mathcal{H}\text{ead}(s) = f_{v_2(I(s))}$ ($v_2(k)$ est la valuation en 2 de k, c'est-à-dire l'exposant de 2 dans la décomposition de k en facteurs premiers).

Interprétation de Narendran et Rusinowitch

On associe à chaque symbole f une indéterminée X_f et on interprète dans le domaine $D = \mathbb{N}[X_f, f \in \mathcal{F}]^X$. I est définie par

- si a est une constante, alors $\phi_a = X_a + 1$;
- si f n'est pas une constante :

$$\phi_f(X_1, \ldots, X_n) = (X_f + 1)\left((X_f^2 + 2X_f)X_1 \cdots X_n + (X_f + 1)(X_1 + \cdots + X_n) + 1\right);$$

- si x est une variable, $I(x) = x$.

Vérifions les quatre propriétés demandées.

(1) Il suffit de vérifier la formule $b^2 = b + ac$.

(2) Supposons que $I(s) = I(t)$ où $s = f(s_1, \ldots, s_n)$ et $t = g(t_1, \ldots, t_n)$. Les cas où f ou g est une constante sont immédiats. Raisonnons par l'absurde et supposons $f \neq g$. Alors on a :

$$\begin{cases} I(s) = (X_f + 1)\left((X_f^2 + 2X_f)I(s_1) \cdots I(s_n) + (X_f + 1)(I(s_1) + \cdots + I(s_n)) + 1\right) \\ I(t) = (X_g + 1)\left((X_g^2 + 2X_g)I(t_1) \cdots I(t_m) + (X_g + 1)(I(t_1) + \cdots + I(t_m)) + 1\right) \end{cases}$$

et comme $I(s) = I(t)$, $X_f + 1$ divise $I(t)$. Comme $X_f + 1$ et $X_g + 1$ sont premiers entre eux, on peut appliquer le lemme de Gauss dans l'anneau factoriel $\mathbb{Z}[X_f, f \in \mathcal{F}_{\mathrm{AC}}]$ ce qui donne

$$X_f + 1 \mid (X_g^2 + 2X_g)I(t_1) \cdots I(t_m) + (X_g + 1)(I(t_1) + \cdots + I(t_m)) + 1$$

Cette propriété reste vraie si on applique une substitution. Appliquons la substitution $\sigma = \{X_g \mapsto X_f, g \in \mathcal{F} - \{f\}\}$. On obtient :

$$X_f + 1 \mid ((X_f^2 + 1)^2 - 1)\sigma(I(t_1) \cdots I(t_m)) + (X_f + 1)\sigma((I(t_1) + \cdots + I(t_m))) + 1$$

et donc

$$X_f + 1 \mid -\sigma(I(t_1)) \cdots \sigma(I(t_m)) + 1$$

mais tout $\sigma(I(t_i))$ possède un facteur $X_f + 1$ donc

$$X_f + 1 \mid 1$$

une contradiction.

(3) Vrai par le lemme 2.25.

(4) Trivial.

Interprétation de Nieuwenhuis et Rubio

On prend $D = \mathcal{M}(TV(\mathcal{F} \cup \{\perp\}, \mathcal{F}_{\mathrm{AC}}, X))$, où $\perp$ est une nouvelle constante, muni de l'extension multi-ensemble d'un ordre *rpo* $\succ_1$ pour une précédence totale arbitraire hormis la condition que $\perp$ soit minimale, avec statut LEX pour les symboles non AC et MUL pour les symboles AC. On définit $I(s)$ par la forme normale, pour la stratégie *leftmost-innermost* et avec aplatissement en priorité, de $\{s\}$ pour le système :

$$\left\{ \begin{array}{l} \{f(u_1, \ldots, u_p, g(s_1, \ldots, s_k), v_1, \ldots v_q)\} \to \{f(u_1, \ldots, u_p, s_i, v_1, \ldots v_q)\}_{i \mid s_i} \text{ max. pour } \succ_1 \\ \{f(u_1, \ldots, u_p, a, v_1, \ldots v_q)\} \to \{f(u_1, \ldots, u_p, \perp, v_1, \ldots v_q)\} \end{array} \right.$$

pour tout symbole AC f et pour tout symbole g qui n'est pas une constante et tel que $f > g$, et pour toute constante a telle que $f > a$.

Vérifions les quatre propriétés demandées.

(1) Évident puisque les termes sont aplatis.

(2) Clair car les règles ne modifient pas le symbole de tête.

(3) La démonstration est relativement longue et technique, voir [76].

(4) Immédiat.

On peut remarquer que cette interprétation ne satisfait pas la propriété sous-terme *strict*.

Exemple. comparons pour cet ordre les deux membres de la règle de distributivité $x*(y+z)$ et $(x*y)+(x*z)$:

$$I(x*(y+z)) = \{x*y, x*z\} \prec I((x*y)+(x*z)) = \{(x*y)+(x*z)\}$$

La première interprétation donne ce résultat car y et z sont incomparables donc tous deux maximaux. On constate ainsi que cet ordre oriente la distributivité dans le sens contraire du sens naturel.

2.5.4 Application à la définition d'ordres AC totaux sur des structures classiques

AC et identité, idempotence, nilpotence

Les règles concernées sont

$$\left\{ \begin{array}{l} x + 0 \to x \\ x + x \to x \\ x + x \to 0 \\ x + 0 \to 0 \end{array} \right.$$

qui, à part la troisième règle, s'orientent correctement grâce à la propriété sous-terme. Pour orienter la troisième règle, le plus simple est d'utiliser l'ordre AC total de Nieuwenhuis et Rubio avec une précédence satisfaisant $+ > 0$. Ceci est d'ailleurs un très bon exemple montrant la difficulté d'utilisation des ordres totaux basés sur les interprétation polynomiales, car on peut vérifier que si l'on désire prouver la terminaison de $x + x \rightarrow 0$ avec un tel ordre total, on est amené à utiliser l'interprétation

$$\begin{cases} I(x+y) = 2(18xy + 4(x+y) + 1) \\ I(0) = 3 \end{cases}$$

avec la deuxième interprétation à valeur dans les entiers que nous avons introduit, la première n'étant pas utilisable car nous ne connaissons pas *a priori* le nombre d'opérateurs de la précédence; et avec l'interprétation de Narendran et Rusinowitch, on a l'interprétation

$$\begin{cases} I(x+y) = (X_+ + 1)((X_+^2 + 2X_+)xy + (X_+ + 1)(x+y) + 1) \\ I(0) = X_0 + 1 \end{cases}$$

et on choisit $X_+ > X_0$.

Groupes abéliens

Ici pas de problème de toute façon car il n'y a qu'un symbole AC. On a déjà vu que l'ordre *rpo* défini par une précédence totale où $- > + > 0$ et tous les autres symboles sont plus grand que $-$ définissait un ordre AC total qui orientait correctement les règles de réécriture des groupes abéliens.

Anneaux commutatifs

Nous avons vu dans la section sur l'ordre *apo* que l'on pouvait orienter les règles du système canonique des anneaux avec l'ordre $(\succ_{apo}, \succ_D)$. Cet préordre est total si la précédence l'est mais les classes de congruences associées sont plus grandes que les classes AC : ce sont les classes ACD. Pour fabriquer un ordre AC total, il suffit alors de compléter l'ordre précédent avec pour troisième composante l'un des ordres totaux définis précédemment par interprétation AC, si l'on veut un ordre basé sur la précédence on peut prendre $(\succ_{apo}, \succ_D, \succ_{nr})$, où $\succ_{nr}$ est n'importe lequel des ordres AC totaux définis précédemment.

2.6 Conclusion sur les ordres sur les termes

En conclusion, nous pouvons dire que les définitions précédentes d'ordres AC totaux, obtenues en 1991 et 1993, et également la technique nouvelle que nous avons donnée permettant d'orienter la distributivité dans le « bon » sens, constituent un progrès considérable, car grâce à de tels ordres nous allons pouvoir montrer au chapitre 5 des résultats de décidabilité de problèmes de mots.

Néanmoins, les résultats obtenus ne sont pas entièrement satisfaisants : d'abord du point de l'implantation, les définitions d'ordres totaux avec plusieurs symboles AC sont très lourdes, à un point tel qu'à notre connaissance, l'ordre de Narendran et Rusinowitch n'a jamais été implanté par personne, et l'ordre de Nieuwenhuis et Rubio non plus (mais ce dernier est très récent). Nous ne les avons d'ailleurs pas implanté non plus car bien que théoriquement il aurait fallu, pour la complétion modulo les anneaux commutatifs, implanter l'ordre à trois composantes précédent, c'est-à-dire l'ordre *apo*, l'ordre polynomial $\succ_D$ et l'ordre de Nieuwenhuis et Rubio; il se trouve que l'on n'en a pas besoin car on ne compare que des termes qui sont déjà en forme normale, donc en particulier il est inutile de distribuer. Il serait intéressant de regarder si ce phénomène, évidemment très particulier au cas des anneaux commutatifs, se généralise en un concept qui serait un *ordre sur les formes normales modulo* S.

Ainsi, bien que les résultats d'existence d'ordres AC totaux soient une avancée considérable, il reste encore aussi bien du point implantation que du point de vue théorique du travail à faire.

Chapitre 3

Complétion associative-commutative variadique

Nous présentons dans ce chapitre une variante de l'algorithme de complétion associative-commutative de Peterson et Stickel [78].

Ici, nous présentons une version légèrement différente : l'algorithme précité utilise les *règles d'extension*. Par l'utilisation des termes aplatis, nous pouvons nous passer de ces règles, à condition de généraliser la notion de réécriture et la notion de paires critiques. Cette idée de l'aplatissement pour se passer des extensions n'est pas nouvelle, mais il est difficile de dire qui en a la paternité, car c'est une idée qui est apparue d'abord en tant que technique d'implantation, les termes variadiques étant naturellement une bonne implantation de l'algèbre des termes en présence de symboles AC. On retrouve par exemple cette notion dans un article de Kapur, Musser et Narendran [50] sur des critères d'élimination de paires critiques; dans un article de Anantharaman, Hsiang et Mzali [3], dans la thèse de Domenjoud [33]. Mais il semble que jusqu'à présent, personne n'ait poussé l'idée au maximum, en se passant des extensions d'un point de vue théorique. L'idée essentielle est que l'on généralise le filtrage, donc la réécriture, de manière à pouvoir dire qu'un système de règles, sans ses extensions, est convergent. Par exemple, on veut pouvoir dire que le système

$$\begin{cases} x + 0 \to x \\ x + (-x) \to 0 \\ -0 \to 0 \\ -(-x) \to x \\ -(x+y) \to (-x) + (-y) \end{cases}$$

c'est-à-dire sans la règle d'extension $x + y + (-y) \to x$ est convergent modulo AC, pour cette notion de « réécriture étendue ».

En fait, cette variante variadique de la réécriture et de la complétion associative-commutative présente plusieurs avantages, tant du point de vue théorique que pratique.

- Avantages théoriques :
  - il n'y a plus besoin de *protéger* les règles d'extension, puisqu'elles n'existent plus, et donc il n'y a pas besoin de séparer l'ensemble des règles que l'on manipule en deux parties, les règles protégées et les règles non protégées;

- l'absence de règles d'extension permet une preuve beaucoup plus élémentaire de la décroissance de la réécriture de preuve;
- lorsque l'ensemble d'équations de départ est sans variables, alors il n'y a pas d'introduction de règles ni d'équations avec variables : cet avantage est essentiel pour la complétion close (chapitre 5);
- on peut démontrer un *résultat d'unicité* de l'ensemble canonique de règles produit par la complétion. Un tel résultat d'unicité à déjà été montré par Jouannaud et Kirchner [43] pour la complétion modulo une théorie arbitraire E à classes finies, mais l'obtention par la complétion d'un tel système n'était obtenu qu'après un traitement spécial en fin de complétion. La difficulté de l'obtention de l'unicité en présence de règles étendues est due à ce que les règles étendues sont simplifiables dans leur membre gauche, et donc on ne peut pas définir la notion d'inter-réduction de façon aussi simple que dans le cas standard. Ici le problème ne se pose plus car on se passe des règles étendues, et l'on peut alors définir l'inter-réduction de la même façon que dans le cas standard, et enfin montrer l'unicité également de la même façon que dans le cas standard.

• Avantages pratiques :

- suppression des traitements particuliers aux extensions, la stratégie en est largement simplifiée;
- un avantage possible qui reste à étudier : utiliser des algorithmes de filtrage et d'unification étendus de manière à calculer en une fois les filtres et les unificateurs, c'est-à-dire pour éviter d'appeler la fonction de filtrage ou d'unification une fois pour une règle et une deuxième fois pour la règle étendue. Nous avons fait des essais dans notre implantation au niveau du filtrage, avec des résultats prometteurs, mais cela n'a pas été terminé par manque de temps.

3.1 Réécriture étendue sur les termes aplatis

Dans cette section, nous étendons la notion de réécriture sur les termes aplatis. Nous avons défini au chapitre 1 les notions de position, de sous-terme et de remplacement pour les termes aplatis.

L'idée de la réécriture aplatie est de remplacer l'utilisation des règles étendues à la Peterson et Stickel par l'utilisation d'un filtrage étendu. Par exemple, on voudrait que la réécriture suivante soit possible :

$$a + b + (-a) \xrightarrow[x+(-x)\to 0]{\Lambda,\{x\mapsto a\}} b + 0$$

Définition 3.1 *Un terme variadique s filtre de façon étendue un terme variadique t s'il existe une substitution σ telle que*

- $t \equiv s\sigma$ *ou*
- $\mathcal{H}\mathrm{ead}(s) = \mathcal{H}\mathrm{ead}(t) = + \in \mathcal{F}_{\mathrm{AC}}$ *et* $t \equiv (s + x)\sigma$ *pour une « nouvelle » variable* x*, c'est-à-dire* $x \notin \mathcal{V}\mathrm{ar}(s)$*.*

On note alors $t \geq_{\mathrm{e}} s$*.*

Nous définissons alors la réécriture étendue de façon naturelle :

Définition 3.2 *Un terme variadique s se réécrit en t à la position p de façon étendue par une règle $l \to r$ si*

$$\begin{cases} s|_p \geq_{\mathrm{e}} l \ (\text{par } \sigma, \text{ c-à-d } s|_p = l\sigma \text{ ou } (l+x)\sigma) \\ t = s[r\sigma]_p \end{cases}$$

ce que l'on note $s \xrightarrow[l \to r]{p,\sigma}_e t$.

Ce qui est intéressant en pratique, c'est de faire une fonction de filtrage qui recherche un filtre en même temps par l et par $l + x$, ce qui gagne du temps.

3.1.1 Terminaison, confluence et propriété de Church-Rosser

Les définitions de terminaison, de confluence et de propriété de Church-Rosser donnée au chapitre 1 sont évidemment applicables à la relation de réécriture aplatie par un système de règles aplaties R. La terminaison peut encore être testée par un ordre AC.

3.1.2 Lemme des paires critiques

Afin de tester la confluence de la relation de réécriture étendue par un système de règles aplaties, nous définissons une notion de paire critique adaptée à la structure de terme aplatie, dite paire critique étendue. Pour faire un parallèle entre notre méthode et la méthode des règles étendues « à la Peterson-Stickel », on peut dire que ici on ne calcule pas les règles étendues mais par contre on est obligé de calculer les paires critiques issues aussi bien des règles elles-mêmes que de leurs éventuelles extensions. Cette idée a été initialement utilisée pour la complétion AC-close [66, 71], et ici nous l'utilisons dans le cas non clos.

Exemple. On voit facilement que les règles $a+b \to c$ et $a+d \to e$ induisent comme conséquence équationnelle $c + d = b + e$:

$$\begin{array}{ccc} & a+b+d & \\ \swarrow_{a+b\to c} & & {}_{a+d\to e}\searrow \\ c+d & & B+e \end{array}.$$

Cette équation est une paire critique étendue entre les deux règles précédentes, qui peut être obtenue en unifiant modulo $\equiv$ les membres gauches étendus $x + a + b$ et $y + a + d$. De la même façon, les deux règles $f(x + a + b) \to x$ et $a + c \to d$ induisent l'équation $f(b + d) = c$ en unifiant le sous-terme du membre gauche de la première règle $x + a + b$ avec le membre gauche étendu $y + a + c$.

Ces deux exemples montrent pourquoi on est amené à la définition suivante.

Définition 3.3 *Soient deux règles aplaties $l_1 \to r_1$ et $l_2 \to r_2$. On dit que ces deux règles se* superposent de façon étendue *en tête si* $\mathcal{H}\mathrm{ead}(l_1) = \mathcal{H}\mathrm{ead}(l_2)$ *et s'il existe une substitution σ telle que*

- $l_1\sigma \equiv l_2\sigma$ *ou,* $\mathcal{H}\mathrm{ead}(l_1) = + \in \mathcal{F}_{\mathrm{AC}}$ *et*

- $(l_1 + x)\sigma \equiv l_2\sigma$ *ou*
- $l_1\sigma \equiv (l_2 + y)\sigma$ *ou*
- $(l_1 + x)\sigma \equiv (l_2 + y)\sigma$,

avec x et y deux « nouvelles » variables, c'est-à-dire ni dans $\mathcal{V}\mathrm{ar}(l_1)$ ni dans $\mathcal{V}\mathrm{ar}(l_2)$. On note ceci $l_1\sigma \equiv_e l_2\sigma$.

On dit que $l_2 \to r_2$ se superpose de façon étendue *à la position p dans $l_1 \to r_1$ ($p \in \mathcal{FP}\mathrm{os}(l_1) - \{\Lambda\}$) si $\mathcal{H}\mathrm{ead}(l_2) = \mathcal{H}\mathrm{ead}(l_1|_p)$ et s'il existe une substitution σ telle que*

- $l_1|_p\sigma \equiv l_2\sigma$ *ou,* $\mathcal{H}\mathrm{ead}(l_2) = + \in \mathcal{F}_{\mathrm{AC}}$ *et*
- $l_1|_p\sigma \equiv (l_2 + y)\sigma$.

À chaque fois, la paire critique *associée est obtenue en appliquant les règles aux instances correspondantes.* L'ensemble des paires critiques étendues *est donc :*

$$
\begin{array}{rcl}
CP_e(l_1 \to r_1, l_2 \to r_2) & = & CP_e^{(\Lambda)}(l_1 \to r_1, l_2 \to r_2) \\
& \cup & \displaystyle\bigcup_{\substack{p \in \mathcal{FP}\mathrm{os}(l_1) \\ p \neq \Lambda}} CP_e^{(p)}(l_1 \to r_1, l_2 \to r_2) \\
& \cup & \displaystyle\bigcup_{\substack{p \in \mathcal{FP}\mathrm{os}(l_2) \\ p \neq \Lambda}} CP_e^{(p)}(l_2 \to r_2, l_1 \to r_1)
\end{array}
$$

où

$$
\begin{array}{rcl}
CP_e^{(\Lambda)}(l_1 \to r_1, l_2 \to r_2) & = & \{r_1\sigma = r_2\sigma\} \\
& \cup & \{r_1\sigma = (r_2 + y)\sigma, (r_1 + x)\sigma = r_2\sigma, (r_1 + x)\sigma = (r_2 + y)\sigma\} \\
& si & \mathcal{H}\mathrm{ead}(l_1) = \mathcal{H}\mathrm{ead}(l_2) = + \in \mathcal{F}_{\mathrm{AC}}
\end{array}
$$

et

$$
\begin{array}{rcl}
CP_e^{(p)}(l_1 \to r_1, l_2 \to r_2) & = & \{r_1\sigma = l_1[r_2]_p\sigma\} \\
& \cup & \{r_1\sigma = l_1[r_2 + y]_p\sigma\} \\
& si & \mathcal{H}\mathrm{ead}(l_1) = \mathcal{H}\mathrm{ead}(l_2) = + \in \mathcal{F}_{\mathrm{AC}}
\end{array}
$$

La proposition suivante est la version étendue du lemme des paires critiques bien connu, qui permet de restreindre la propriété de confluence à un ensemble fini de cas, qui sont justement donnés par l'ensemble des paires critiques étendues.

Proposition 3.4 *Soit R un ensemble de règles aplaties. La relation de réécriture étendue $\underset{R}{\longrightarrow}_e$ est localement confluente si et seulement si toutes les paires critiques étendues de R sont joignables.*

Démonstration. Il faut montrer que tous les pics commutent ou se joignent par une paire critique. Considérons une preuve en pic

$$s \xleftarrow[l\to r]{\sigma,p} t \xleftrightarrow[\equiv]{*} t' \xrightarrow[g\to d]{\sigma,pq} u$$

Si les étapes $\equiv$ sont au-dessus de pq, on peut appliquer des règles de commutation de la figure 3.1. Sinon, il y a deux cas :

- si $q \notin \overline{\mathcal{FP}\mathrm{os}}(l)$, on peut appliquer une règle de commutation;

Commutation d'une falaise, cas de positions parallèles

$$s \xleftrightarrow[\equiv]{\sigma,p} t \xrightarrow[l_1\to r_1]{\sigma,q} u \Longrightarrow s \xrightarrow[l_1\to r_1]{\sigma,q} v \xleftrightarrow[\equiv]{\sigma,p} u$$
$$\text{si } p \parallel q$$

Commutation d'une falaise, cas d'une superposition dans une variable

$$s \xleftrightarrow[\equiv]{\sigma,p} t \xrightarrow[l_1\to r_1]{\sigma,pq} u \Longrightarrow s \xrightarrow[l_1\to r_1]{\sigma,o} v \xleftrightarrow[\equiv]{\sigma',p} u$$
$$\text{où } Level(q) \neq \Lambda$$

Commutation d'un pic, cas de positions parallèles

$$s \xleftarrow[l_1\to r_1]{\sigma,p} t \xrightarrow[l_2\to r_2]{\sigma,q} u \Longrightarrow s \xrightarrow[l_2\to r_2]{\sigma,q} v \xleftarrow[l_1\to r_1]{\sigma,p} u$$
$$\text{si } p \parallel q$$

Commutation d'un pic, cas d'une superposition dans une variable

$$s \xleftarrow[l_1\to r_1]{\sigma,p} t \xleftrightarrow[\equiv]{(\geq pq)*} t' \xrightarrow[l_2\to r_2]{\sigma,pq} u \Longrightarrow$$
$$s \left(\xleftrightarrow[\equiv]{*} \quad \xrightarrow[l_2\to r_2]{\sigma,o} \right)_{o\in P} s' \xleftarrow[l_1\to r_1]{\sigma',p} v \xleftrightarrow[\equiv]{*} u' \left(\xleftarrow[l_2\to r_2]{\sigma,o'} \quad \xleftrightarrow[\equiv]{*} \right)_{o'\in P'} u$$
$$\text{où} \begin{cases} q \notin \mathcal{FP}\mathrm{os}(v_1) \\ q = p'q' \text{ avec } r_1|_{p'} = x \in X \\ P = \{o \in \mathcal{P}\mathrm{os}(r_1) | r_1|_o = x\} \\ P' = \{o' \in \mathcal{P}\mathrm{os}(l_1) | l_1|_{o'} = x\} \\ \sigma' : \begin{cases} x\sigma' = x\sigma[r_2\sigma]_{q'} \\ \text{si } y \neq x,\ y\sigma' = y\sigma \end{cases} \end{cases}$$

Figure 3.1 : Réécriture de preuve : règles de commutation

- si $q \in \overline{\mathcal{FP}\mathrm{os}}(l)$, on a une superposition : il existe une paire critique $u = v \in CP_e^q(l|_q, g)$ telle que

$$s \underset{\equiv}{\overset{*}{\longleftrightarrow}} s' \underset{u=v}{\overset{\sigma',p}{\longleftrightarrow}} u' \underset{\equiv}{\overset{*}{\longleftrightarrow}} u$$

□

3.2 Complétion Associative-Commutative variadique

Dans cette section, nous donnons l'algorithme de complétion modulo AC sur les règles aplaties. La majeure partie de la section est consacrée à la démonstration du théorème fondamental, le théorème de complétude.

Orientation

$$E \cup \{u = v\}; R \vdash E; R \cup \{u \to v\}$$
$$\text{si } u \succ v$$

Déduction

$$E; R \vdash E \cup \{u = v\}; R$$
$$\text{si } u = v \in CP_e(R)$$

Effacement

$$E \cup \{u = v\}; R \vdash E; R$$
$$\text{si } u \equiv v$$

Composition

$$E; R \cup \{u \to v\} \vdash E; R \cup \{u \to v'\}$$
$$\text{si } v \underset{R}{\longrightarrow}_e v'$$

Simplification

$$E \cup \{u = v\}; R \vdash E \cup \{u' = v\}; R$$
$$\text{si } u \underset{R}{\longrightarrow}_e u'$$

Réduction

$$E; R \cup \{u \to v\} \vdash E \cup \{u' = v\}; R$$
$$\text{si } \begin{cases} l \to r \in R \\ u \underset{l \to r}{\overset{\theta,p}{\longrightarrow}}_e u' \\ p \neq \Lambda \text{ ou} \\ p = \Lambda \text{ et } \theta \text{ n'est pas un renommage ou} \\ p = \Lambda,\ \theta \text{ renommage et } v \succ r\theta \end{cases}$$

Figure 3.2 : Règles d'inférence de la complétion associative-commutative variadique

3.2.1 Règles d'inférence de la complétion associative-commutative variadique

Dans la suite, $\succ$ est un ordre AC, E est l'ensemble des équations, R est l'ensemble des règles.

La figure 3.2 donne les règles d'inférence de la complétion . La règle ORIENTATION transforme une équation en règle, la règle DÉDUCTION ajoute des paires critiques comme nouvelles équations, EFFACEMENT supprime les équations triviales, COMPOSITION réduit les membres droits de règles, SIMPLIFICATION réduit les équations, RÉDUCTION réduit les membres gauches de règles.

Un algorithme de *complétion AC variadique* est un algorithme de complétion basé sur cet ensemble de règles, au sens de la définition 1.46.

3.2.2 Équité et complétude

Définition 3.5 *Une dérivation $E_0; R_0 \vdash E_1; R_1 \vdash \ldots$ est* équitable *si toutes les paires critiques persistantes sont calculées, c'est-à-dire :*

$$CP_e(R_\infty) \subset \bigcup_{i=0}^{\infty} E_i.$$

Un algorithme de complétion est équitable si toutes les dérivations qu'il peut produire sont équitables.

L'équité est une propriété fondamentale, car c'est elle qui assure la complétude de la stratégie. En pratique, nous voulons également simplifier les règles autant que possible : cela conduit à un ensemble de règles inter-réduit, une propriété importante puisqu'elle permet d'obtenir un résultat d'unicité de l'ensemble de règles produit.

Théorème 3.6 *Si la complétion est équitable et n'échoue pas alors*

$$\forall s,t \quad s =_{E_0 \cup \mathrm{AC}} t \Longleftrightarrow \exists u,v \quad \begin{tikzcd} s \arrow[dr, "*", "R_{\infty,e}"'] & & & t \arrow[dl, "*"', "R_{\infty,e}"] \\ & u \equiv & v & \end{tikzcd}.$$

3.2.3 Démonstration de la complétude

La démonstration de la complétude se fait par la méthode désormais classique de normalisation des preuves [7]. Elle se fait en cinq étapes :

- définition de l'algèbre des preuves formelles;
- définition d'un système de réécriture des preuves, dérivé des règles de commutation, et des règles d'inférence;
- preuve de la correction de ce système;
- preuve de la terminaison de ce système;
- conclusion.

Algèbre des preuves formelles

Définition 3.7 *Les preuves élémentaires sont*

$$\text{(étapes AC)} \quad s \overset{*}{\underset{\equiv}{\longleftrightarrow}} t \quad \text{si} \quad \left\{ s \equiv t \right.$$

$$\text{(étape équationnelle)} \quad s \overset{\sigma,p}{\underset{l=r}{\longleftrightarrow}} t \quad \text{si} \quad \left\{ \begin{array}{l} s|_p = l\sigma \\ t = s[r\sigma]_p \end{array} \right.$$

$$\text{(étape de réécriture)} \quad s \overset{\sigma,p}{\underset{l\to r}{\longrightarrow}} t \quad \text{si} \quad \left\{ \begin{array}{l} s|_p = l\sigma \\ t = s[r\sigma]_p \end{array} \right.$$

ainsi que la preuve symétrique pour l'étape de réécriture. Une preuve est une suite finie de preuves élémentaires : étant données deux preuves P et Q, $P.Q$ est une preuve si et seulement si last(P) =first(Q) où last et first sont données par :

$$\left\{ \begin{array}{l} \mathit{first}(s\ldots t) = s \text{ et } \mathit{last}(s\ldots t) = t \text{ si } s\ldots t \text{ est une preuve élémentaire} \\ \mathit{first}(PQ) = \mathit{first}(P) \text{ et } \mathit{last}(PQ) = \mathit{last}(Q) \end{array} \right.$$

Une preuve P est une $E; R$-preuve si toute équation (resp. règle) qui apparaît dans ses sous-preuves élémentaires est dans E (resp. R).

Notre but va être de montrer que toute preuve d'égalité entre deux termes, constituée uniquement d'étapes AC et d'étapes équationnelles, peut être transformée en une preuve par réécriture AC. L'idée est d'une part d'éliminer les étapes équationnelles en les transformant en étapes de réécriture par orientation, et d'autre part d'éliminer les portions de preuve de la forme $\leftarrow\ \ \rightarrow$, que nous appelons des *pics*. De plus, pour assurer que la réécriture utilisée est bien la réécriture AC-étendue variadique et pas la réécriture modulo AC générale, il faut aussi éliminer les preuves de la forme $\overset{p}{\underset{\equiv}{\longleftrightarrow}}\ \overset{q}{\rightarrow}$ où la position p n'est pas suffixe de la position q, nous appelons ces preuves des *falaises* (voir figure 3.3).

Règles de réécriture de preuves

Le système de réécriture de preuves est constitué des règles de commutation de la figure 3.1 et des règles déduites des règles d'inférence de la figure 3.4.

Correction du système

Il s'agit de vérifier que les règles de réécriture de preuves reflètent bien les inférences.

Lemme 3.8 *Si $E; R \vdash E'; R'$ et si P est une $E; R$-preuve alors $P \underset{E\cup E';R\cup R'}{\Longrightarrow} Q$ où Q est une $E'; R'$-preuve.*

Démonstration. Il faut vérifier la propriété pour chacune des règles d'inférence. Cela ne pose pas de problème théorique, c'est uniquement technique. La règle qui pose le plus de difficultés est DÉDUCTION, les transformations de preuve dans ce cas-là étant issues du lemme du paires critiques étendues (proposition 3.4). □

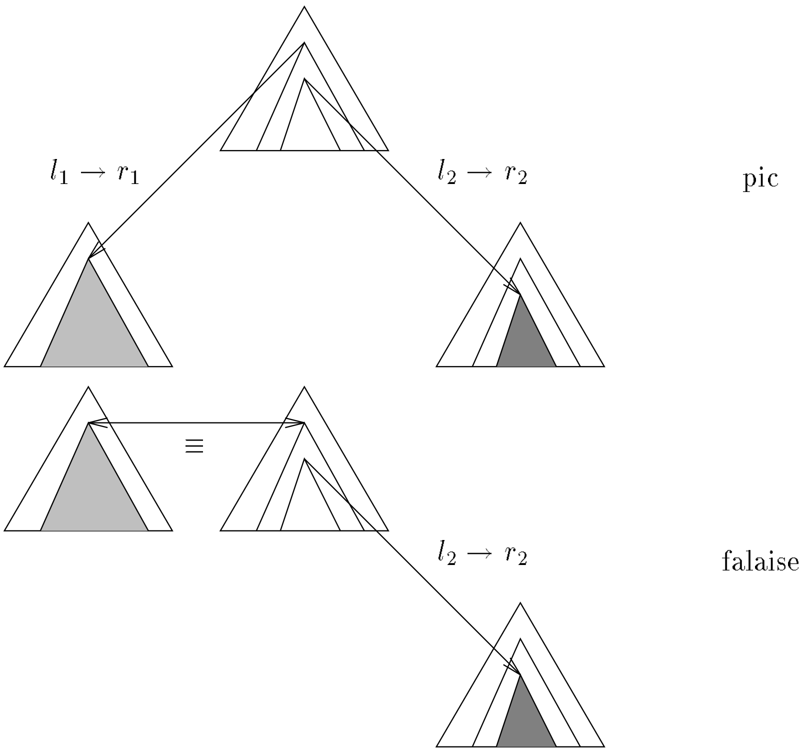

Figure 3.3 : Une preuve en pic et une preuve en falaise

Terminaison de la réécriture de preuves

La complexité d'une preuve est le multi-ensemble des complexités de ses sous-preuves élémentaires, définie par :

$$\begin{array}{rcllll}
C(s \underset{\equiv}{\longleftrightarrow} t) & = & \langle \{s\}, & \bot, & \bot & \rangle \\
C(s \underset{l=r}{\longleftrightarrow} t) & = & \langle \{s,t\}, & \bot, & \bot & \rangle \\
C(s \overset{\sigma}{\underset{l\to r}{\longrightarrow}} t) & = & \langle \{s\}, & l, & r\sigma & \rangle
\end{array}$$

où $\bot$ est un nouvel élément, minimal pour tous les ordres. Deux complexités élémentaires sont comparées par l'extension lexicographique des ordres $\succ_{mul}$ pour la première composante, $\trianglerighteq_{\mathrm{AC}}$ pour la deuxième et $\succ$ pour la troisième. Les preuves sont alors comparées par l'extension multi-ensemble de l'ordre ci-dessus.

Cet ordre est bien-fondé puisqu'il est construit à partir d'ordres bien-fondés (l'ajout de l'élément minimal $\bot$ ne pose pas de problème) par extension par les fonctionnelles lexicographique et multi-ensemble.

Il nous faut maintenant vérifier que chaque règle décroît pour cet ordre. Ces vérifications sont énumérées figure 3.5.

Orientation

$$s \overset{\sigma,p}{\underset{l=r}{\longleftrightarrow}} t \Longrightarrow s \overset{\sigma,p}{\underset{l\to r}{\longrightarrow}} t$$
$$\text{si } l \succ r$$

Déduction

$$s \overset{\sigma,pA}{\underset{l\to r}{\longleftarrow}} t \overset{*}{\underset{\equiv}{\longleftrightarrow}} t' \overset{\sigma,pqB}{\underset{g\to d}{\longrightarrow}} u \Longrightarrow s \overset{*}{\underset{\equiv}{\longleftrightarrow}} s' \overset{\sigma'}{\underset{u=v}{\longleftrightarrow}} u' \overset{*}{\underset{\equiv}{\longleftrightarrow}} u$$
$$\text{si } \begin{cases} q \in \mathcal{FP}\text{os}(l) \\ u = v \in CP_e(l|_q, g) \end{cases}$$

Effacement

$$s \overset{\sigma,p}{\underset{l=r}{\longleftrightarrow}} t \Longrightarrow s \overset{*}{\underset{\equiv}{\longleftrightarrow}} t$$
$$\text{si } l \equiv r$$

Composition

$$s \overset{\sigma,p}{\underset{l\to r}{\longrightarrow}} t \Longrightarrow s \overset{\sigma,p}{\underset{l\to r'}{\longrightarrow}} u \overset{\theta\sigma,pq}{\underset{g\to d}{\longleftarrow}} t' \overset{*}{\underset{\equiv}{\longleftrightarrow}} t$$
$$\text{si } r \overset{\theta,q}{\underset{g\to d/\equiv}{\longrightarrow}} r'$$

Simplification

$$s \overset{\sigma,p}{\underset{l=r}{\longleftrightarrow}} t \Longrightarrow s \overset{*}{\underset{\equiv}{\longleftrightarrow}} s' \overset{\theta\sigma,pq}{\underset{g\to d}{\longrightarrow}} u \overset{\sigma,p}{\underset{l'=r}{\longleftrightarrow}} t$$
$$\text{si } l \overset{\theta,q}{\underset{g\to d/\equiv}{\longrightarrow}} l'$$

Réduction

$$s \overset{\sigma,p}{\underset{l\to r}{\longrightarrow}} t \Longrightarrow s \overset{*}{\underset{\equiv}{\longleftrightarrow}} s' \overset{\theta\sigma,pq}{\underset{g\to d}{\longrightarrow}} u \overset{\sigma,p}{\underset{l'=r}{\longleftrightarrow}} t$$
$$\text{si } \begin{cases} l \overset{\theta,q}{\underset{g\to d/\equiv}{\longrightarrow}} l' \\ p \neq \Lambda \text{ ou} \\ p = \Lambda \text{ et } \theta \text{ n'est pas un renommage ou} \\ p = \Lambda,\ \theta \text{ renommage et } u \succ r\theta \end{cases}$$

Figure 3.4 : Réécriture de preuve : règles reflètant les règles d'inférence

règle $P \Longrightarrow Q$	$C(P)$	$C(Q)$	preuve de $P \succ_P Q$
COMMUTATION DES PICS EN PARALLÈLE	$\langle\{t\},\ldots\rangle$, $\langle\{t\},\ldots\rangle$	$\langle\{s\},\ldots\rangle$, $\langle\{u\},\ldots\rangle$	$t \succ s$ et $t \succ u$
COMMUTATION DES PICS ANCÊTRES	$\langle\{t\},\ldots\rangle$, $\langle\{t\},\ldots\rangle$	$\langle\{s\},\ldots\rangle$, $\langle\{u\},\ldots\rangle$	$t \succ s$ et $t \succ u$
COMMUTATION DES FALAISES EN PARALLÈLE	$\langle\{t\},\ldots\rangle$, $\langle\{t\},\ldots\rangle$	$\langle\{s\},\ldots\rangle$, $\langle\{u\},\ldots\rangle$	$t \succ s$ et $t \succ u$
COMMUTATION DES FALAISES ANCÊTRES	$\langle\{t\},\ldots\rangle$, $\langle\{t\},\ldots\rangle$	$\langle\{s\},\ldots\rangle$, $\langle\{u\},\ldots\rangle$	$t \succ s$ et $t \succ u$
ORIENTATION	$\langle\{s,t\},\ldots\rangle$	$\langle\{s\},\ldots\rangle$	$\{s,t\} \succ_{mul} \{s\}$
DÉDUCTION	$\langle\{t\},\ldots\rangle$, $\langle\{t\},\ldots\rangle$, $\langle\{t'\},\ldots\rangle$	$\langle\{s\},\ldots\rangle$, $\langle\{s',u'\},\ldots\rangle$, $\langle\{u'\},\ldots\rangle$	$t \succ s \equiv s'$ et $t \succ u \equiv u'$
EFFACEMENT	$\langle\{s,t\},\ldots\rangle$	$\langle\{s\},\ldots\rangle$	$\{s,t\} \succ_{mul} \{s\}$
COMPOSITION	$\langle\{s\},l\sigma,r\sigma\rangle$	$\langle\{s\},l\sigma,r'\sigma\rangle$, $\langle\{t'\},\ldots\rangle$, $\langle\{t'\},\ldots\rangle$	$r \succ r'$ et $s \succ t \equiv t'$
SIMPLIFICATION	$\langle\{s,t\},\ldots\rangle$	$\langle\{s\},\ldots\rangle$, $\langle\{s'\},\ldots\rangle$, $\langle\{u,t\},\ldots\rangle$	$s \succ t \succ u$
RÉDUCTION	$\langle\{s\},l\sigma,r\sigma\rangle$	$\langle\{s\},\perp,\perp\rangle$, $\langle\{s'\},g\theta\sigma,d\theta\sigma\rangle$, $\langle\{u,t\},\ldots\rangle$	$l\sigma \succ \perp$, $(l,r) \succ (g\theta,d\theta)$, $s \equiv s' \succ u$ et $s \succ t$

Figure 3.5 : Terminaison de la réécriture des preuves

Conclusion de la démonstration de la complétude

La relation $\Longrightarrow$ schématise une classe de systèmes de réécriture $\underset{E;R}{\Longrightarrow}$, c'est-à-dire, pour tous E et R, $P \underset{E;R}{\Longrightarrow} Q$ si P et Q sont des $E;R$-preuves et $P \Longrightarrow Q$. Nous pouvons maintenant démontrer la complétude. Posons

$$\mathcal{E}_i = \bigcup_{j=0}^{i} E_j \text{ et } \mathcal{R}_i = \bigcup_{j=0}^{i} R_j.$$

Une suite d'inférences à partir de $E_0; R_0$ peut être reflétée au niveau des preuves dans le sens suivant : une $E_0; R_0$-preuve peut être réécrite par $\underset{\mathcal{E}_0;\mathcal{R}_0}{\Longrightarrow}$ en donnant une $E_0; R_0$-preuve, qui est aussi une $\mathcal{E}_1; \mathcal{R}_1$-preuve donc peut être à son tour réécrite par $\underset{\mathcal{E}_1;\mathcal{R}_1}{\Longrightarrow}$, etc...Si P est une $\mathcal{E}_i; \mathcal{R}_i$-preuve, soit $P\downarrow_i$ une forme normale de P pour $\underset{\mathcal{E}_i;\mathcal{R}_i}{\Longrightarrow}$ (existe car ce système termine), qui est donc également une $\mathcal{E}_i; \mathcal{R}_i$-preuve.

Lemme 3.9 *$P\downarrow_i$ est une $E_i; R_i$-preuve.*

Démonstration. par l'absurde. Si ce n'est pas une $E_i; R_i$-preuve, elle contient une étape élémentaire utilisant une équation (resp. une règle) dans un E_j (resp. R_j) pour un $j < i$ qui n'est pas dans E_i (resp. R_i). Mais quelle que soit l'étape d'inférence qui a supprimé cette équation (resp. règle), il existe une règle de réécriture de preuve correspondante pouvant être appliquée, donc la preuve n'est pas en forme normale. □

Supposons maintenant que $s =_{E_0 \cup AC} t$. Alors il existe une $E_0; R_0$-preuve P telle que $s = \mathrm{first}(P)$ et $t = \mathrm{last}(P)$. La suite d'inférences produite par la complétion est reflétée en une séquence de preuves

$$P_0 = P\downarrow_0 \overset{*}{\Longrightarrow} P_1 = P_0\downarrow_1 \overset{*}{\Longrightarrow} P_2 = P_1\downarrow_2 \overset{*}{\Longrightarrow} \cdots$$

Puisque $\Longrightarrow$ termine, cette suite est stationnaire, donc il existe n tel que

$$P_n = P_{n+1} = P_{n+2} = \cdots$$

Par le lemme précédent, P_n est une $E_k; R_k$-preuve pour tout $k \geq n$, c'est donc une $E_\infty; R_\infty$-preuve. La complétion n'échouant pas, E_∞ est vide, donc P_n contient uniquement des étapes de réécriture et des étapes AC. Puisqu'elle est en forme normale pour $\underset{\mathcal{E}_n;\mathcal{R}_n}{\Longrightarrow}$, les seuls pics qu'elle peut posséder sont critiques (sinon une règle de commutation s'applique). Mais par l'hypothèse d'équité, les paires critiques associées à ces pics ont été calculées à une certaine étape j. Si $j < n$, alors P_n n'est pas en forme normale pour $\underset{\mathcal{E}_n;\mathcal{R}_n}{\Longrightarrow}$ ce qui n'est pas possible, et si $j \geq n$, alors P_n n'est pas en forme normale pour $\underset{\mathcal{E}_j;\mathcal{R}_j}{\Longrightarrow}$, ce qui est également contradictoire. Par conséquent, P_n ne possède aucun pic, et est donc de la forme

$$\begin{tikzcd}[column sep=small] s \arrow[dr, "*", "R_{\infty,e}"'] & & & t \arrow[dl, "*"', "R_{\infty,e}"] \\ & u \equiv & v & \end{tikzcd}.$$

3.3 L'exemple standard de complétion associative-commutative : les groupes abéliens

La théorie des groupes abéliens est l'exemple le plus classique de complétion AC. On part de l'ensemble d'équations

$$\begin{cases} AC(+) \\ x + 0 = x \\ x + (-x) = 0 \end{cases}$$

Source	Script de la complétion

`operators`
`  + : AC`
`  0 : constant`
`  - : unary`
`  x,y : variable`
 	
`theory empty`
 	
`axioms`
`  x+0 = x;`
`  x+-(x) = 0;`
 	
`order  rpo ->+>0 ; + mul`
 	
`end`
	`Resultat :`
	`{x+-(x) -> 0,`
	`-(-(x)) -> x,`
	`x+0 -> x,`
	`-(0) -> 0,`
	`-(x+y) -> -(y)+-(x)} (5 regles)`
	
	`Nombre d'appels au filtrage AC     : 9517`
	`Nombre d'appels reussis            : 800 (8%)`
	`Nombre d'appels a l'unification AC : 39`
	`Nombre de solutions trouvees       : 242 (6.21 en moyenne)`
	`Temps utilisateur : 21.070 s`
	`Temps systeme : 0.620 s`

Figure 3.6 : Complétion des groupes abéliens modulo AC

.

et on prend comme ordre l'ordre *rpo* avec la précédence $- > + > 0$ et le statut multi-ensemble pour $+$. Ceci est bien un ordre AC.

Le fichier source pour la complétion des groupes abéliens dans notre implantation et le résultat obtenu sont donnés figure 3.6.

On trouvera d'autres exemples de complétion AC, ainsi qu'une description détaillée de l'implantation, dans l'annexe A.

Il faut noter que l'on trouve une règle de moins que dans le système canonique obtenue par une complétion à la Peterson-Stickel. La règle manquante est $x + (-x) + y \to y$, qui est l'extension de $x + (-x) \to 0$, et bien sûr cette règle n'est pas présente ici puisque notre méthode n'utilise pas les extensions. La différence tient dans le fait que notre système est convergent pour la réécriture aplatie, alors que le système à six règles de Peterson et Stickel est convergent pour la réécriture AC-étendue.

3.4 Unicité du système canonique

Nous montrons maintenant un résultat d'unicité du système canonique produit par la complétion AC variadique. Comme nous n'avons pas de règles d'extensions, nous pouvons définir l'inter-réduction de la même façon que dans le cas standard : R est inter-réduit si pour toute règle $l \to r$ de R, r est R-irréductible, et l est $R - \{l \to r\}$-irréductible.

Théorème 3.10 *Soit $\succ$ un ordre de simplification compatible avec AC et soient R_1 et R_2 deux systèmes canoniques modulo AC définissant la même théorie équationnelle (modulo AC), et qui soient tous deux inclus dans $\succ$ ($\underset{R_i}{\longrightarrow} \subseteq \succ$). Alors R_1 et R_2 sont égaux modulo AC à renommage près.*

Démonstration. soit $l \to r$ un règle de R_1. On a

$$l \overset{*}{\underset{R_2 \cup \mathrm{AC}}{\longleftrightarrow}} r$$

donc comme R_2 est convergent modulo AC, il existe u et v tels que

$$l \xrightarrow[R_2/\mathrm{AC}]{*} u \xleftrightarrow[\mathrm{AC}]{*} v \xleftarrow[R_2/\mathrm{AC}]{*} r.$$

On a $l \succ r \succeq u$ donc la réécriture de l en u possède au moins une étape : il existe une règle $g \to d$ dans R_2 telle que

$$l \xrightarrow[g\to d/\mathrm{AC}]{\sigma,p} \xrightarrow[R_2/\mathrm{AC}]{*} u,$$

et en particulier $l \trianglerighteq_{\mathrm{e}} g$. Par le même raisonnement appliqué à $g \to d$ et en échangeant les rôles de R_1 et R_2, on trouve qu'il existe une règle $l' \to r'$ telle que $g \trianglerighteq_{\mathrm{e}} l'$. Mais comme R_1 est inter-réduit, on ne peut pas avoir $l \vartriangleright_{\mathrm{e}} l'$ donc $l \equiv g\sigma$ et σ est un renommage. On a donc une réécriture de la forme

$$l \xrightarrow[g\to d/\mathrm{AC}]{} d\sigma \xrightarrow[R_2/\mathrm{AC}]{*} u,$$

mais $d\sigma$ n'est pas R_2-réductible car R_2 est inter-réduit (et σ est un renommage), donc $d\sigma =_{\mathrm{AC}} u$. D'autre part, si on suppose que la réécriture de r en u se fait en au moins une étape, on déduit aisément que r est R_1-réductible ce qui est impossible, donc $r =_{\mathrm{AC}} u$. On a donc montré que $l =_{\mathrm{AC}} g\sigma$ et $r =_{\mathrm{AC}} d\sigma$. Ainsi pour toute règle de R_1 il existe une règle de R_2 égale modulo AC à renommage près, et vice-versa puisque R_1 et R_2 jouent un rôle symétrique. □

Chapitre 4

Réécriture et complétion normalisées

Dans ce chapitre, nous introduisons la notion de réécriture normalisée. La réécriture S-normalisée, pour une théorie équationnelle S, est une modification de la réécriture modulo S, mais qui suppose la théorie S donnée par un système convergent (éventuellement modulo AC, en présence de symboles AC). Pour montrer la terminaison de la réécriture modulo E par un système de règles R, il faut tester la décroissance des règles de R par un ordre compatible avec E. Ceci pose un gros problème car on ne sait pas en général construire de tels ordres. Nous montrons que pour tester la terminaison de cette réécriture S-normalisée, il suffit d'un ordre compatible avec AC, et AC seulement, pas avec la théorie S toute entière.

Ensuite, nous nous intéressons à la complétion. Pour cela, nous étudions en détail quelles sont les réductions de preuves que l'on désire avoir, selon la technique de Devie [32], et nous en déduisons les règles d'inférences pour la complétion normalisée. La règle d'orientation consiste alors à fabriquer à partir d'une équation non pas *une* règle, mais un *ensemble* de règles, dites *règles normalisées*, ainsi que d'un *ensemble d'équations*, dites *équations critiques*. Quant à la règle de déduction, elle reste presque identique, si ce n'est que l'unification se fait modulo une théorie qui peut être choisie arbitrairement, pourvu qu'elle contienne la théorie AC et soit contenue dans la théorie définie par $S \cup \mathrm{AC}$. Ce choix va nous permettre de trouver un bon compromis entre deux objectifs contradictoires :

1. choisir une théorie pas trop « grosse » pour que l'unification modulo cette théorie soit décidable et finitaire (on ne choisira donc pas d'unifier modulo ACD);

2. choisir une théorie pas trop « petite » pour avoir un petit nombre d'unificateurs (comme on l'a déjà remarqué, l'unification modulo AC1, ACI ou AC1I est plus efficace que modulo AC).

Bien sûr, on peut toujours, au pire, se contenter d'unifier modulo AC dans le cas général.

Nous donnons alors une expression générale (c'est-à-dire valable quelle que soit la théorie S) des règles normalisées et des équations critiques à calculer, et nous donnons également ensuite des expressions « optimisées » pour des théories particulières, pour lesquelles on montre le gain en efficacité du processus de complétion. Ces optimisations s'appliquent en particulier au cas des polynômes sur un anneau commutatif, et la complétion normalisée se trouve alors être un algorithme de calcul de bases standard d'idéaux de polynômes, analogue aux algorithmes de Buchberger et de Kandri-Rody et Kapur.

4.1 Notion de réécriture normalisée

Dans cette section nous introduisons une nouvelle notion de réécriture : la réécriture S-normalisée, où S est un système de règles convergent modulo AC. Il s'agit en fait de réécrire uniquement les termes en forme normale pour S.

4.1.1 Définition de la réécriture S-normalisée

Supposons donné un ensemble de règles S, convergent modulo AC. Tout terme s possède donc une forme normale unique (modulo AC) pour S, que l'on note $s\downarrow_S$.

Définition 4.1 *La* relation de réécriture S-normalisée*, notée* $s \xrightarrow[l\to r/S]{\sigma,p} t$*, est définie par :*

$$\begin{cases} s' = s\downarrow_S \\ s' \xrightarrow[l\to r]{\sigma,p}_e t \end{cases}$$

Exemple. si $S = \mathrm{AC1}(+,0)$, définie par le système d'une règle $\{x + 0 \to x\}$, convergent modulo AC, alors on a par exemple la réduction

$$-(a+b) \xrightarrow[-(x+y)\to(-x)+(-y)/S]{} (-a)+(-b)$$

mais par contre la réduction suivante n'est pas valide

$$-(0+b) \xrightarrow[-(x+y)\to(-x)+(-y)/S]{} (-0)+(-b)$$

parce que la forme normale de $-(0+b)$ modulo AC1 est $(-b)$, qui n'est pas une instance de $-(x+y)$. On remarque, sur cet exemple, qu'en fait dans le cas de AC1, la notion de réécriture normalisée coïncide avec la notion de réécriture AC1-contrainte [45]. Le lecteur se convaincra facilement que c'est toujours vrai.

4.1.2 Terminaison de la réécriture S-normalisée

La terminaison de la réécriture est, contrairement au cas de la notion habituelle de réécriture modulo, très facile à tester. Supposons donné un ordre de réduction AC-compatible $\geq_{\mathrm{AC}}$ tel que $\to_S \subseteq >_{\mathrm{AC}}$, c'est-à-dire tel que les règles de S décroissent pour $\geq_{\mathrm{AC}}$. On a alors la proposition suivante.

Proposition 4.2 *Soit R un ensemble de règles tel que $\forall l \to r \in R$, $l >_{\mathrm{AC}} r$. Alors la relation de réécriture S-normalisée par R est noethérienne.*

Démonstration. si $l >_{\mathrm{AC}} r$ et $s \xrightarrow[l\to r/S]{} t$ alors, si $s' = s\downarrow_S$, $s' \xrightarrow[l\to r]{}_e t$ donc on a $s' >_{\mathrm{AC}} t$ mais $s \geq_{\mathrm{AC}} s'$ donc $s >_{\mathrm{AC}} t$. □

4.2 Transformation des preuves et couples normalisants

Nous allons donner dans la prochaine section l'ensemble des règles d'inférence de la complétion S-normalisée. La complétude sera démontrée par la méthode de normalisation des preuves. En fait, c'est en étudiant quelles sont les transformations de preuves dont nous avons besoin que l'on définit les règles d'inférence, et ceci de manière quasi-automatique comme l'a remarqué Hervé Devie [32]. Aussi, nous allons prendre explicitement ce point de vue : nous allons d'abord déterminer l'ensemble des réécritures de preuves dont nous avons besoin pour transformer n'importe quelle preuve équationnelle en une preuve par réécriture normalisée, et seulement ensuite nous en déduirons les règles d'inférence nécessaires par l'algorithme de complétion.

Afin de montrer que toute preuve peut être réécrite en une preuve par réécriture normalisée, nous allons construire des règles de réécriture de preuves pour transformer n'importe quelle étape équationnelle, ainsi que n'importe quel « pic » : cela suffit pour que toute preuve en forme normale soit de la forme

$$s \xrightarrow[R/S]{*} s' \xleftrightarrow[\equiv]{*} t' \xleftarrow[R/S]{*} t$$

La démarche que nous allons suivre est la suivante : nous définissons d'abord l'algèbre des preuves dont nous avons besoin, tenant compte de la réécriture normalisée, puis nous étudions ensuite en détail de quelle façon on peut transformer une étape équationnelle, c'est-à-dire de la forme

$$s \xleftrightarrow[u=v]{\sigma,p} t$$

Dans cette étude, nous sommes amené à considérer de nouvelles équations, dites *équations critiques* pour pourvoir pourvoir transformer certains types de preuves. Nous regardons ensuite comment on peut transformer les preuves en forme de pic, c'est-à-dire de la forme

$$s \xleftarrow[l_1 \to r_1]{} t \xleftrightarrow[S \cup \mathrm{AC}]{*} t' \xrightarrow[l_2 \to r_2]{} u$$

Nous verrons dans la suite que nous aurons besoin de « choisir » parmi différentes variantes possibles de définition des équations critiques. Nous sommes amené ainsi à définir la notion de *couple S-normalisant*, pour lequel on a besoin de définir la notion de complexité des preuves. C'est pourquoi la démarche consiste à définir la complexité, puis de façon générale la notion de couple S-normalisant, et enfin de donner différentes définitions valides de couple S-normalisant, la première valide quelle que soit S, puis les suivantes valides pour certains S particuliers, et possédant de meilleures propriétés que la définition générale.

4.2.1 Algèbre des preuves

Définition 4.3 *Les preuves élémentaires sont*

$$\text{(étapes AC)} \quad s \xleftrightarrow[\equiv]{(\geq p)^*} t \quad si \quad \begin{cases} s|_p \equiv t|_p \\ t = s[t|_p]_p \end{cases}$$

$$\text{(étape équationnelle)} \quad s \xleftrightarrow[l=r]{\sigma,p} t \quad si \quad \begin{cases} s|_p \equiv l\sigma \\ t = s[r\sigma]_p \end{cases}$$

$$\text{(étape de réécriture par } S\text{)} \quad s \xrightarrow[l\to r]{\sigma,p}_S t \quad si \quad \begin{cases} l \to r \in S \\ s|_p \equiv l\sigma \\ t = s[r\sigma]_p \end{cases}$$

$$\text{(étape de réécriture)} \quad s \xrightarrow[l\to r]{\sigma,p} t \quad si \quad \begin{cases} s|_p \equiv l\sigma \\ t = s[r\sigma]_p \end{cases}$$

ainsi que les symétriques pour les preuves élémentaires par réécriture. Les preuves sont alors des séquences de preuves élémentaires, définies comme au chapitre 3.

4.2.2 Réécriture des étapes équationnelles

Dans cette section, nous allons montrer comment on réécrit n'importe quelle étape de preuve de la forme

$$s \xleftrightarrow[u=v]{\sigma,p} t.$$

Pour cela, nous devons faire attention qu'avant d'utiliser une étape de réécriture, il nous faut regarder si le terme à réécrire est en S-forme normale, sinon ce ne sera pas une réécriture S-normalisée.

On se fixe un ordre de réduction AC $\succeq$.

1. Si u ou v est S-réductible. Soit $u' = u\downarrow_S$ et $v' = v\downarrow_S$. On peut réécrire la preuve en

$$s \xrightarrow{(\geq p)*}_S s' \xleftrightarrow[u'=v']{\sigma,p} t' \xleftarrow{(\geq p)*}_S t$$

2. Si σ est S-réductible. On peut réécrire la preuve en

$$s \xrightarrow{(\geq p)*}_S s' \xleftrightarrow[u=v]{\sigma',p} t' \xleftarrow{(\geq p)*}_S t$$

   où $\sigma' = \sigma\downarrow_S$.

3. Supposons dorénavant que u, v et σ sont S-irréductibles.

   (a) Si $u =_{\mathrm{AC}} v$. Alors la preuve devient trivialement

$$s \xleftrightarrow[\equiv]{*} t.$$

(b) Si $u \succ v$ (par exemple, le cas $v \succ u$ est similaire).

i. Si s est S-irréductible, alors on peut transformer la preuve en

$$s \xrightarrow[u \to v]{\sigma, p} t.$$

ii. Sinon, $s \xrightarrow[l \to r]{q, \sigma} s'$ pour une certaine règle $l \to r$ de S. Nous discutons suivant les positions p et q.

A. Si $p \parallel q$, nous pouvons normaliser la preuve en

$$s \underset{l \to r}{\overset{q,\sigma}{\xrightarrow{S}}} s' \xleftrightarrow[u=v]{p,\sigma} t' \underset{l \to r}{\overset{q,\sigma}{\xleftarrow{S}}} t.$$

B. Si $p = qp'$ avec $p' \notin \mathcal{FP}\mathrm{os}(l)$. Nous pouvons normaliser sous la forme

$$s \underset{l \to r}{\overset{q,\sigma}{\xrightarrow{S}}} s' \left(\xleftrightarrow[u=v]{o',\sigma} \right)^*_{o' \in P'} s'' \underset{l \to r}{\overset{q,\sigma'}{\xleftarrow{S}}} t' \left(\xleftrightarrow[u=v]{o,\sigma} \right)^*_{o \in P} t,$$

où

$$\left\{ \begin{array}{l} p' = q'p'' \text{ avec } l|_{q'} = x \in X \\ P = \{o \in \mathcal{P}\mathrm{os}(l) | l|_o = x\} - \{q'\} \\ P' = \{o' \in \mathcal{P}\mathrm{os}(r) | r|_{o'} = x\} \\ \sigma' : \left\{ \begin{array}{l} x\sigma' = x\sigma[v\sigma]_{p''} \\ \text{si } y \neq x, \; y\sigma' = y\sigma \end{array} \right. \end{array} \right.$$

(voir figure 4.1).

C. Le cas $q = pq'$ avec $q' \notin \mathcal{FP}\mathrm{os}(u)$ n'est pas possible car cela implique que σ est S-réductible.

D. Si $q = pq'$ avec $q' \in \mathcal{FP}\mathrm{os}(u)$, alors $u|_{q'}\sigma =_{\mathrm{AC}} s|_p|_{q'}\sigma =_{\mathrm{AC}} s|_q\sigma =_{\mathrm{AC}} l\sigma$ donc $\exists \theta \in \mathrm{CSU}_{\equiv}(u|_{q'}, l) \exists \lambda \mid \sigma =_{\equiv} \theta\lambda$. θ est S-irréductible puisque σ est S-irréductible. On peut réécrire en la preuve

$$s \underset{l \to r}{\overset{\sigma,q}{\xrightarrow{S}}} s' \xleftrightarrow[\equiv]{*} s'' \xleftrightarrow[u\theta[r\theta]_{q'} = v\theta]{\lambda, p} t' \xleftrightarrow[\equiv]{*} t$$

E. Enfin, si $p = qp'$ avec $p' \in \mathcal{FP}\mathrm{os}(l)$ et $p' \neq \Lambda$. alors $l|_{p'}\sigma =_{\mathrm{AC}} s|_{qp'}\sigma =_{\mathrm{AC}} u\sigma$ donc $\exists \theta \in \mathrm{CSU}_{\equiv}(l|_{p'}, u) \exists \lambda \mid \sigma =_{\equiv} \theta\lambda$. θ est encore S-irréductible, et on peut réécrire en

$$s \underset{l \to r}{\overset{\sigma,q}{\xrightarrow{S}}} s \xleftrightarrow[\equiv]{*} s'' \xleftrightarrow[r\theta = l\theta[v\theta]_{p'}]{\lambda, q} t' \xleftrightarrow[\equiv]{*} t$$

(c) Dernière possibilité, si u et v ne sont pas comparable par $\succeq$, alors l'équation $u = v$ ne peut pas s'orienter, nous sommes dans un cas d'échec. En fait, la complétion échouera que s'il n'y a pas moyen de se débarrasser autrement de cette équation, c'est-à-dire sauf si cette équation peut être simplifiée.

L'étude précédente conduit à un premier ensemble de règles de réécriture de preuves, les règles dites de normalisation (voir figure 4.2).

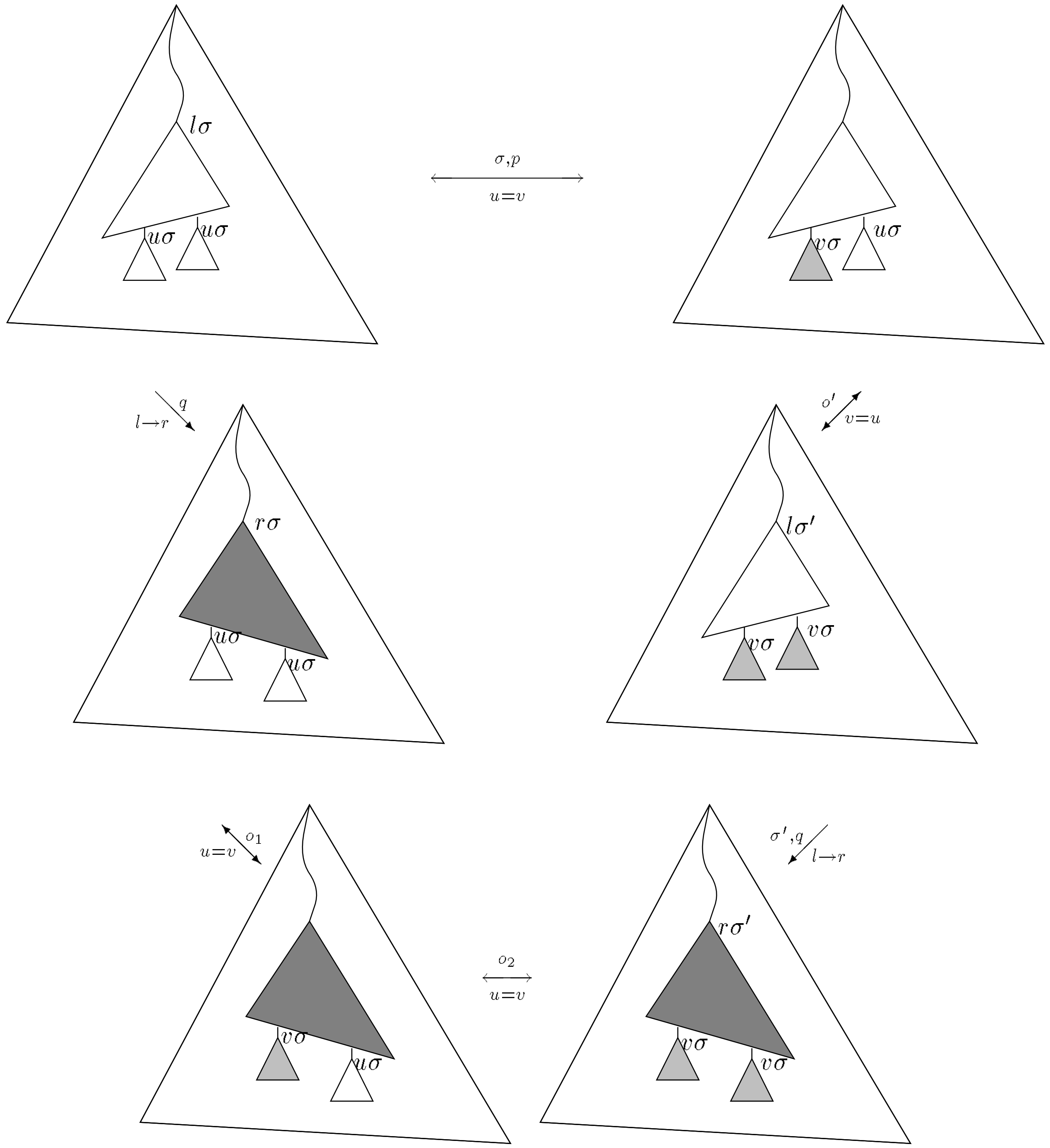

Figure 4.1 : S-normalisation d'une étape équationnelle à une position ancêtre

Normalisation des substitutions

$$s \overset{\sigma,p}{\underset{u=v}{\longleftrightarrow}} t \Longrightarrow s \overset{(\geq p)*}{\underset{S}{\longrightarrow}} s' \overset{\sigma',p}{\underset{u=v}{\longleftrightarrow}} t' \overset{(\geq p)*}{\underset{S}{\longleftarrow}} t$$
$$\text{si } \sigma' = \sigma\downarrow_S$$

Normalisation des équations

$$s \overset{\sigma,p}{\underset{u=v}{\longleftrightarrow}} t \Longrightarrow s \overset{(\geq p)*}{\underset{S}{\longrightarrow}} s' \overset{\sigma,p}{\underset{u'=v'}{\longleftrightarrow}} t' \overset{(\geq p)*}{\underset{S}{\longleftarrow}} t$$
$$\text{si } u' = u\downarrow_S \text{ et } v' = v\downarrow_S$$

Normalisation à une position parallèle

$$s \overset{\sigma,p}{\underset{u=v}{\longleftrightarrow}} t \Longrightarrow s \overset{q,\sigma}{\underset{l\to r}{\xrightarrow{S}}} s' \overset{p,\sigma}{\underset{u=v}{\longleftrightarrow}} t' \overset{q,\sigma}{\underset{l\to r}{\xleftarrow{S}}} t$$
$$\text{si } l \to r \in S \text{ et } p \parallel q$$

Normalisation à une position ancêtre

$$s \overset{\sigma,p}{\underset{u=v}{\longleftrightarrow}} t \Longrightarrow s \overset{p,\sigma}{\underset{l\to r}{\xrightarrow{S}}} s' \left(\overset{o,\sigma}{\underset{u=v}{\longleftrightarrow}} \right)^*_{o'\in P'} s'' \overset{p,\sigma'}{\underset{l\to r}{\xleftarrow{S}}} t' \left(\overset{o',\sigma}{\underset{u=v}{\longleftrightarrow}} \right)^*_{o\in P} t$$
$$\text{si } l \to r \in S,\ p = qp' \text{ avec } \begin{cases} p' \notin \mathcal{FP}\mathrm{os}(l) \\ p' = q'p'' \text{ avec } l|_{q'} = x \in \mathcal{X} \\ P = \{o \in \mathcal{P}\mathrm{os}(l) | l|_o = x\} - \{q'\} \\ P' = \{o' \in \mathcal{P}\mathrm{os}(r) | r|_{o'} = x\} \\ \sigma' : \begin{cases} x\sigma' = x\sigma[v\sigma]_{p''} \\ \text{si } y \neq x,\ y\sigma' = y\sigma \end{cases} \end{cases}$$

Figure 4.2 : Réécriture de preuves : règles de normalisation par S

4.2.3 Réécriture des pics

Voyons maintenant quelles sont les transformations de preuve nécessaires pour transformer les pics. Nous avons tout fait jusqu'à présent pour ne pas introduire d'étape de réécriture pour lesquelles le terme que l'on réécrit n'est pas en forme normale pour S. Nous pourrions donc supposer qu'aucune des deux réécritures du pic ne se fait par une règle de S. Malheureusement, nous allons voir que dans les règles qui vont suivre, on est obligé d'introduire de tels pics, et donc nous devons aussi considérer ce cas.

Les pics entre deux règles non dans S se traitent de la même manière que dans le cas AC standard : soit les positions sont parallèles ou ancêtres, auquel cas on va appliquer une commutation, soit il va falloir introduire une paire critique par unification. Les cas de commutation sont standard (voir figure 4.3), néanmoins il faut remarquer qu'ils peuvent effectivement

S-CONFLUENCE

$$s \xleftarrow[S]{+} t \xleftrightarrow[\equiv]{*} t' \xrightarrow[S]{+} u \Longrightarrow s \xrightarrow[S]{*} s' \xleftrightarrow[\equiv]{*} u' \xleftarrow[S]{*} u$$

COMMUTATION DES PICS EN PARALLÈLE

$$s \xleftarrow[l_1 \to r_1]{\sigma,p} t \xrightarrow[l_2 \to r_2]{\sigma,q} u \Longrightarrow s \xrightarrow[l_2 \to r_2]{\sigma,q} v \xleftarrow[l_1 \to r_1]{\sigma,p} u$$
$$\text{si } p \parallel q$$

COMMUTATION DES PICS SE SUPERPOSANT DANS UNE VARIABLE

$$s \xleftarrow[l_1 \to r_1]{\sigma,p} t \xrightarrow[l_2 \to r_2]{\sigma,pq} u \Longrightarrow$$
$$s \left(\xleftrightarrow[\equiv]{*} \quad \xrightarrow[l_2 \to r_2]{\sigma,o} \right)_{o \in P} s' \xleftarrow[l_1 \to r_1]{\sigma',p} v \xleftrightarrow[\equiv]{*} u' \left(\xleftarrow[l_2 \to r_2]{\sigma,o'} \quad \xleftrightarrow[\equiv]{*} \right)_{o' \in P'} u$$
$$\text{où} \left\{ \begin{array}{l} q \notin \mathcal{FP}\mathrm{os}(v_1) \\ q = p'q' \text{ avec } r_1|_{p'} = x \in X \\ P = \{o \in \mathcal{P}\mathrm{os}(r_1) | r_1|_o = x\} \\ P' = \{o' \in \mathcal{P}\mathrm{os}(l_1) | l_1|_{o'} = x\} \\ \sigma' : \left\{ \begin{array}{l} x\sigma' = x\sigma[r_2\sigma]_{q'} \\ \text{si } y \neq x,\ y\sigma' = y\sigma \end{array} \right. \end{array} \right.$$

Figure 4.3 : Réécriture de preuves : transformation des pics non critiques

introduire des réécritures pour lesquelles on ne sait si le terme que l'on réécrit est en forme normale pour S, et donc il va nous falloir ensuite donner des règles de réécriture pour de telles preuves. Dans le cas critique on peut remarquer une chose intéressante : on a le choix de la théorie modulo laquelle on unifie : on peut choisir d'unifier modulo AC, ce qui donnera une preuve transformée de la forme

$$\xleftrightarrow[\equiv]{*} \; \xleftrightarrow[CP_{\mathrm{AC}}]{} \; \xleftrightarrow[\equiv]{*}$$

ou on peut choisir d'unifier modulo $\mathrm{AC} \cup S$, ce qui donnera une preuve de la forme

$$\searrow^{*}_{S} \; \xleftrightarrow[\equiv]{*} \; \swarrow^{*}_{S} \; \xleftrightarrow[CP_{\mathrm{AC} \cup S}]{} \; \searrow^{*}_{S} \; \xleftrightarrow[\equiv]{*} \; \swarrow^{*}_{S}$$

La seule difficulté est dans le deuxième cas d'assurer que la preuve ainsi produite est plus petite que la preuve de départ. Cela sera résolu par l'introduction du terme $c(s, p, t)$ dans la mesure de complexité. En fait, il est correct et complet de calculer les paires critiques par unification modulo n'importe quelle théorie T contenant AC et contenue dans $\mathrm{AC} \cup S$. Ceci

Normalisation à une position parallèle

$$s \xrightarrow[u\to v]{\sigma,p} t \Longrightarrow s \underset{l\to r}{\overset{q,\sigma}{\longrightarrow_S}} s' \xrightarrow[u\to v]{p,\sigma} t' \underset{l\to r}{\overset{q,\sigma}{{}_S\longleftarrow}} t$$
$$\text{si } l \to r \in S \text{ et } p \parallel q$$

Normalisation à une position ancêtre

$$s \xrightarrow[u\to v]{\sigma,p} t \Longrightarrow s \underset{l\to r}{\overset{p,\sigma}{\longrightarrow_S}} s' \left(\xrightarrow[u\to v]{o,\sigma}\right)^*_{o'\in P'} s'' \underset{l\to r}{\overset{p,\sigma'}{{}_S\longleftarrow}} t' \left(\xleftarrow[u\to v]{o',\sigma}\right)^*_{o\in P} t$$
$$\text{si } l \to r \in S,\ p = qp' \text{ avec } \begin{cases} p' \notin \mathcal{FP}\mathrm{os}(l) \\ p' = q'p'' \text{ avec } l|_{q'} = x \in \mathcal{X} \\ P = \{o \in \mathcal{P}\mathrm{os}(l) | l|_o = x\} - \{q'\} \\ P' = \{o' \in \mathcal{P}\mathrm{os}(r) | r|_{o'} = x\} \\ \sigma' : \begin{cases} x\sigma' = x\sigma[v\sigma]_{p''} \\ \text{si } y \neq x,\ y\sigma' = y\sigma \end{cases} \end{cases}$$

Figure 4.4 : Réécriture de preuves : transformation des réécritures non S-normalisée

va nous permettre d'utiliser les unifications plus efficaces que AC, comme AC1 et ACI, que l'on a vu au chapitre 1.

L'ensemble des paires critiques entre deux règles $l_1 \to r_1$ et $l_2 \to r_2$, sera donc donné par :

$$\begin{array}{rcl} CP_T(l_1 \to r_1, l_2 \to r_2) & = & CP_T^{(\Lambda)}(l_1 \to r_1, l_2 \to r_2) \\ & \cup & \displaystyle\bigcup_{\substack{p \in \mathcal{FP}\mathrm{os}(l_1) \\ p \neq \Lambda}} CP_T^{(p)}(l_1 \to r_1, l_2 \to r_2) \\ & \cup & \displaystyle\bigcup_{\substack{p \in \mathcal{FP}\mathrm{os}(l_2) \\ p \neq \Lambda}} CP_T^{(p)}(l_2 \to r_2, l_1 \to r_1) \end{array}$$

où $CP_T^{(\Lambda)}(l_1 \to r_1, l_2 \to r_2)$ contient, pour l'ensemble des T-unificateurs σ :

$$\begin{array}{ll} & \{r_1\sigma = r_2\sigma\} \\ \cup & \{r_1\sigma = (r_2 + y)\sigma, (r_1 + x)\sigma = r_2\sigma, (r_1 + x)\sigma = (r_2 + y)\sigma\} \\ \text{si} & \mathcal{H}\mathrm{ead}(l_1) = \mathcal{H}\mathrm{ead}(l_2) = + \in \mathcal{F}_{\mathrm{AC}} \end{array}$$

et $CP_T^{(p)}(l_1 \to r_1, l_2 \to r_2)$ contient, pour l'ensemble des T-unificateurs σ :

$$\begin{array}{ll} & \{r_1\sigma = l_1[r_2]_p\sigma\} \\ \cup & \{r_1\sigma = l_1[r_2 + y]_p\sigma\} \\ \text{si} & \mathcal{H}\mathrm{ead}(l_1) = \mathcal{H}\mathrm{ead}(l_2) = + \in \mathcal{F}_{\mathrm{AC}} \end{array}$$

4.2.4 Étapes de réécriture non normalisées

Nous venons de voir que les règles de réécriture des pics peuvent introduire des étapes de réécriture non normalisée. Il nous faut donc des règles pour « normaliser » ces étapes. Pour cela, nous utilisons les mêmes règles que pour transformer les étapes équationnelles non normalisées, en les considérant comme si elles était des équations (voir figure 4.4).

Le point un peu délicat est de savoir comment transformer les cas critiques. Pour cela, nous devrons assurer que pour toute règle $l \to r$, provenant d'une équation $u = v$, les cas critiques d'application de la règle $l \to r$ peuvent se transformer en une preuve utilisant une instance critique de $u = v$. Cela se traduira par une condition d'inclusion sur les couples normalisants.

4.2.5 Complexité des preuves

On utilisera l'abréviation $s\downarrow_p = s[(s|_p)\downarrow_S]_p$, et l'on note $c(s,p,t) = \{s\}$ si $s\downarrow_p = s$, sinon $\{s\downarrow_p, t\downarrow_p\}$. Remarquons que l'on a toujours $\{s\downarrow_p, t\downarrow_p\} \succeq_{mul} c(s,p,t)$.

La complexité d'une preuve est le multi-ensemble des complexités de ses sous-preuves élémentaires, définies par :

$$
\begin{array}{rcllll}
C(s \longleftarrow_{\mathrm{AC}}\longrightarrow t) & = & \langle \bot, & \{s\}, & \bot, & \bot \rangle \\
C(s \overset{\sigma,p}{\underset{l=r}{\longleftrightarrow}} t) & = & \langle \{s\downarrow_p, t\downarrow_p\}, & \{s,t\}, & \bot, & \bot \rangle \\
C(s \overset{\sigma}{\underset{l\to r}{\longrightarrow}} t) & = & \langle c(s,p,t), & \{s\}, & l, & r\sigma \rangle \\
C(s \longrightarrow_S\longrightarrow t) & = & \langle \bot, & \{s\}, & \bot, & \bot \rangle
\end{array}
$$

où $\bot$ est un nouvel élément, minimal pour tous les ordres. Deux complexités élémentaires sont comparées par l'extension lexicographique des ordres $\succ_{mul}$ pour les première et deuxième composantes, $\trianglerighteq_{\mathrm{AC}}$ pour la troisième et $\succ$ pour la quatrième. Les preuves sont alors comparées par l'extension multi-ensemble de l'ordre ci-dessus.

Cet ordre est bien-fondé puisqu'il est construit à partir d'ordres bien-fondés (l'ajout de l'élément minimal $\bot$ ne pose pas de problème) par extension par les fonctionnelles lexicographique et multi-ensemble.

4.2.6 Couples S-normalisants

Définition 4.4 *Une fonction qui à deux termes u et v associe un couple $(\Theta(u,v), \Psi(u,v))$, où $\Theta(u,v)$ est un ensemble d'équations et $\Psi(u,v)$ un ensemble de règles, est dit* couple S-normalisant *(pour l'ordre $\succ$) si*

- *pour toute preuve élémentaire de la forme*

$$s \overset{\sigma}{\underset{u=v}{\longleftrightarrow}} t \text{ ou } s \overset{\sigma}{\underset{u\to v}{\longleftrightarrow}} t$$

avec u, v, σ en forme normale pour S, $u \succ v$, et telle qu'aucune règle de normalisation de la figure 4.2 ne s'applique, il existe une preuve entre s et t utilisant des étapes équationnelles avec des équations de $\Theta(u,v)$, des étapes de réécriture avec des règles de $\Psi(u,v)$, et des étapes de réécriture par S, qui soit plus petite pour la mesure de complexité donnée précédemment;

INSTANCE CRITIQUE D'ÉQUATION, DU PREMIER TYPE

$$s \xleftrightarrow[u=v]{\sigma,p} t \Longrightarrow s \xrightarrow[l\to r]{\sigma,q} s' \xleftrightarrow[\equiv]{*} s'' \xleftrightarrow[u\theta[r\theta]_{q'}=v\theta]{\lambda,p} t' \xleftrightarrow[\equiv]{*} t$$

si $s \succ t$, $q = pq'$ avec $q' \in \mathcal{FP}\mathrm{os}(u)$, σ S-irréductible,

$u|_{q'}\sigma \equiv s|_p|_{q'}\sigma \equiv s|_q\sigma \equiv l\sigma$ donc $\exists\theta \in \mathrm{CSU}_e(u|_{q'}, l)\exists\lambda \mid \sigma \equiv \theta\lambda$

INSTANCE CRITIQUE DU SECOND TYPE

$$s \xleftrightarrow[u=v]{\sigma,p} t \Longrightarrow s \xrightarrow[l\to r]{\sigma,q} s' \xleftrightarrow[\equiv]{*} s'' \xleftrightarrow[r\theta=l\theta[v\theta]_{p'}]{\lambda,q} t' \xleftrightarrow[\equiv]{*} t$$

si $s \succ t$, $s|_p$ S-irréductible, $p = qp'$ avec $p' \in \mathcal{FP}\mathrm{os}(l)$ et $p' \neq \Lambda$, σ S-irréductible

$l|_{p'}\sigma \equiv s|_{qp'}\sigma \equiv u\sigma$ donc $\exists\theta \in \mathrm{CSU}_e(l|_{p'}, u)\exists\lambda \mid \sigma \equiv \theta\lambda$

INSTANCE CRITIQUE DE RÈGLE, DU PREMIER TYPE

$$s \xrightarrow[u\to v]{\sigma,p} t \Longrightarrow s \xrightarrow[l\to r]{\sigma,q}_S s' \xleftrightarrow[\equiv]{*} s'' \xleftrightarrow[u\theta[r\theta]_{q'}=v\theta]{\lambda,p} t' \xleftrightarrow[\equiv]{*} t$$

si $s \succ t$, $q = pq'$ avec $q' \in \mathcal{FP}\mathrm{os}(u)$, σ S-irréductible,

$u|_{q'}\sigma \equiv s|_p|_{q'}\sigma \equiv s|_q\sigma \equiv l\sigma$ donc $\exists\theta \in \mathrm{CSU}_e(u|_{q'}, l)\exists\lambda \mid \sigma \equiv \theta\lambda$

INSTANCE CRITIQUE DE RÈGLE, DU SECOND TYPE

$$s \xrightarrow[u\to v]{\sigma,p} t \Longrightarrow s \xrightarrow[l\to r]{\sigma,q}_S s' \xleftrightarrow[\equiv]{*} s'' \xleftrightarrow[r\theta=l\theta[v\theta]_{p'}]{\lambda,q} t' \xleftrightarrow[\equiv]{*} t$$

si $s \succ t$, $s|_p$ S-irréductible, $p = qp'$ avec $p' \in \mathcal{FP}\mathrm{os}(l)$ et $p' \neq \Lambda$, σ S-irréductible

$l|_{p'}\sigma \equiv s|_{qp'}\sigma \equiv u\sigma$ donc $\exists\theta \in \mathrm{CSU}_e(l|_{p'}, u)\exists\lambda \mid \sigma \equiv \theta\lambda$

Figure 4.5 : Règles de réécriture de preuves : instances critiques

- *pour toute règle $l \to r$ de $\Psi(u, v)$*

$$\begin{cases} \Theta(l, r) \subseteq \Theta(u, v) \\ \Psi(l, r) \subseteq \Psi(u, v) \end{cases}.$$

Les conditions d'inclusion finales sont là pour assurer que l'on peut transformer les réécritures critiques de la même façon que les étapes équationnelles critiques, comme on l'a vu précédemment.

L'étude générale précédente pour transformer les preuves équationnelles nous conduit à la définition suivante de couples normalisants pour une théorie S arbitraire.

Définition 4.5 *Soit*

$$\Theta_{gen}(u, v) \quad = \quad \{u\theta[r\theta]_q = v\theta \mid q \in \mathcal{FP}\mathrm{os}(u), l \to r \in S, \theta \in CSU_{\mathrm{AC}}(u|_q, l)\}$$

règle $P \Longrightarrow Q$	$C(P)$	$C(Q)$	preuve $P \succ_P Q$
ORIENTATION	$\langle\{s\downarrow_p, t\downarrow_p\}, \ldots\rangle$	$\langle\{s\}, \ldots\rangle$	$s = s\downarrow_S = s\downarrow_p$.
INSTANCE CRITIQUE D'ÉQUATION, DU PREMIER TYPE	$\langle\{s\downarrow_p, t\downarrow_p\}, \{s, t\}\ldots\rangle$	$(\langle\bot, \ldots\rangle)^+$, $\langle\{s''\downarrow_p, t'\downarrow_p\}, \{s'', t'\}\ldots\rangle$	$s\downarrow_p \equiv s''\downarrow_p$, $t \equiv t'$, $s \succ s''$.
INSTANCE CRITIQUE D'ÉQUATION, DU SECOND TYPE	$\langle\{s\downarrow_p, t\downarrow_p\}, \{s, t\}\ldots\rangle$	$(\langle\bot, \ldots\rangle)^+$, $\langle\{s''\downarrow_p, t'\downarrow_p\}, \{s'', t'\}\ldots\rangle$	$s\downarrow_p = s \succeq s''\downarrow_p$, $t \equiv t'$, $s \succ s''$.
INSTANCE CRITIQUE DE RÈGLE, DU PREMIER TYPE	$\langle\{s\downarrow_p, t\downarrow_p\}, \{s\}\ldots\rangle$	$(\langle\bot, \ldots\rangle)^+$, $\langle\{s''\downarrow_p, t'\downarrow_p\}, \{s'', t'\}\ldots\rangle$	$s\downarrow_p \equiv s''\downarrow_p$, $s \succ s''$, $s \succ t \equiv t'$.
INSTANCE CRITIQUE DE RÈGLE, DU SECOND TYPE	$\langle\{s\}, \{s\}\ldots\rangle$	$(\langle\bot, \ldots\rangle)^+$, $\langle\{s''\downarrow_p, t'\downarrow_p\}, \{s'', t'\}, \ldots\rangle$	$s\downarrow_p = s \succeq s''\downarrow_p$, $s \succ t \equiv t'$, $s \succ s''$.

Figure 4.6 : Décroissance des preuves dans les cas critiques, pour le couple normalisant général

$$\cup\{l\theta[v\theta]_q = r\theta \mid q \in \mathcal{FP}\mathrm{os}(l), q \neq \Lambda, l \to r \in S, \theta \in CSU_{\mathrm{AC}}(u, l|_q)\}$$

et soit $\Psi_{gen}(u, v) = \{u \to v\}$.

Proposition 4.6 *Le couple* $(\Theta_{gen}(u, v), \Psi_{gen}(u, v))$ *est* S*-normalisant quelle que soit la théorie* S.

Démonstration. Découle de l'étude générale précédente. Les règles de transformations de preuve dans les cas critiques sont données figure 4.5. La condition d'inclusion est triviale. Il reste à vérifier la décroissance de la complexité, ce qui est fait figure 4.6. Il faut remarquer que pour assurer la décroissance, on doit faire l'hypothèse que les règles du premier type s'appliquent prioritairement à celle du second type. □

Cela est évident pour l'instant car $\Psi(u, v) = \{u \to v\}$, mais cela ne sera pas évident pour les définitions « optimisées » de Θ et Ψ.

Exemple. Supposons $S = \mathrm{AC1} = \{z + 0 \to z\}$. Calculons $\Theta_{gen}(-(x + y), (-x) + (-y))$: il faut unifier modulo AC les termes $x + y$ et $z + 0$. On trouve quatre unificateurs principaux :

$$\begin{cases} x \mapsto v_1 \\ y \mapsto 0 \\ z \mapsto v_1 \end{cases} \begin{cases} x \mapsto 0 \\ y \mapsto v_1 \\ z \mapsto v_1 \end{cases} \begin{cases} x \mapsto v_1 \\ y \mapsto v_2 + 0 \\ z \mapsto v_1 + v_2 \end{cases} \begin{cases} x \mapsto v_1 + 0 \\ y \mapsto v_2 \\ z \mapsto v_1 + v_2 \end{cases}$$

Les deux derniers sont S-réductible donc ne sont pas à prendre en considération. Par conséquent $\Theta_{gen}(-(x+y),(-x)+(-y))$ contient deux équations :

$$\begin{cases} (-x) = (-x) + (-0) \\ (-y) = (-0) + (-y) \end{cases}$$

On constate sur cet exemple que l'on retrouve exactement *l'ensemble des instances interdites* associé à l'équation $-(x+y) = (-x) + (-y)$ dans la complétion AC1 contrainte [45].

4.3 Complétion normalisée

Nous donnons à présent l'algorithme de complétion S-normalisée.

4.3.1 Règles d'inférence de la complétion S-normalisée

Dans la suite, $\succ$ est un ordre AC, E est l'ensemble des équations, R est l'ensemble des règles.

La figure 4.7 donne les règles d'inférence de la complétion S-normalisée générique. La règle ORIENTATION remplace une équation par ses équations critiques et ses règles normalisées, la règle DÉDUCTION ajoute des paires critiques comme nouvelles équations, EFFACEMENT supprime les équations triviales, COMPOSITION réduit les membres droits de règles, SIMPLIFICATION réduit les équations, RÉDUCTION réduit les membres gauches de règles.

Un algorithme de *complétion S-normalisée* est un algorithme de complétion basé sur cet ensemble de règles, au sens de la définition 1.46.

4.3.2 Équité et complétude

Définition 4.7 *Une dérivation $E_0; R_0 \vdash E_1; R_1 \vdash \ldots$ est* équitable *si toutes les paires critiques persistantes sont calculées, c'est-à-dire :*

$$CP_T(R_\infty) \subset \bigcup_{i=0}^{\infty} E_i.$$

Un algorithme de complétion est équitable *si toutes les dérivations qu'il peut produire sont équitables.*

Théorème 4.8 *Si la complétion est équitable et n'échoue pas alors*

$$\forall s,t \quad s =_{E_0\cup \mathrm{AC}\cup S} t \iff \exists u,v \quad \begin{tikzcd} s \arrow[dr, "*", "R_\infty/S"'] & & & t \arrow[dl, "*"', "R_\infty/S"] \\ & u & =_{\mathrm{AC}\cup S} \quad v & \end{tikzcd}$$

La preuve de complétude est, comme dans le cas AC, une variante de la méthode générale de normalisation des preuves [7]. Nous avons déjà défini l'algèbre des preuves, ainsi que la mesure de complexité, il nous reste à définir les règles de réécritures reflétant les règles d'inférences (figure 4.8) puis à vérifier que chacune des règles de réécriture de preuves décroît pour cette mesure. La complétude se démontre alors par le même raisonnement qu'au chapitre 3 pour la complétion AC.

ORIENTATION
$$E \cup \{u = v\}; R \vdash E \cup \Theta(u, v); R \cup \Psi(u, v)$$
$$\text{si } u = u\downarrow_S,\ v = v\downarrow_S,\ u \succ v$$

DÉDUCTION
$$E; R \vdash E \cup \{u = v\}; R \text{ si } u = v \in CP_T(R)$$

NORMALISATION
$$E \cup \{u = v\}; R \vdash E \cup \{u\downarrow_S = v\downarrow_S\}; R$$

EFFACEMENT
$$E \cup \{u = v\}; R \vdash E; R \text{ si } u =_{\mathrm{AC}} v$$

COMPOSITION
$$E; R \cup \{u \to v\} \vdash E; R \cup \{u \to v'\} \text{ si } v \xrightarrow[R/S]{} v'$$

SIMPLIFICATION
$$E \cup \{u = v\}; R \vdash E \cup \{u' = v\}; R \text{ si } u \xrightarrow[R/S]{} u'$$

RÉDUCTION
$$E; R \cup \{u \to v\} \vdash E \cup \{u' = v\}; R \text{ si } \begin{cases} l \to r \in R \\ u \xrightarrow[l \to r/S]{\theta, p} u' \\ p \neq \Lambda \text{ ou} \\ p = \Lambda \text{ et } \theta \text{ n'est pas un renommage ou} \\ p = \Lambda,\ \theta \text{ renommage et } u \succ r\theta \end{cases}$$

Figure 4.7 : Règles d'inférence de la complétion S-normalisée

Les vérifications de terminaison sont énumérées sur les figures 4.9, 4.10 et figures 4.11, il faut remarquer que dans plusieurs cas de règles $P \Longrightarrow Q$ où $C(Q)$ contient un $c(s, p, t)$ sans que l'on sache si l'on est dans le cas $s = s\downarrow_p$ ou non, on fait la preuve avec $\{s\downarrow_p, t\downarrow_p\}$, qui comme on l'a déjà remarqué est plus grand que $c(s, p, t)$ dans tous les cas.

4.3.3 Résultat d'unicité et relation avec le système AC-convergent de la même théorie

Une idée qui peut venir à l'esprit concernant l'unicité est de dire que si R est un système canonique pour la réécriture S-normalisée, alors $R \cup AC$ est AC-canonique. Malheureusement, ce n'est pas toujours vrai, comme l'indique la proposition suivante.

Proposition 4.9 *Si R est un système canonique pour la réécriture S-normalisée, et si aucune règle de S n'est réductible par une règle de R alors $R \cup S$ est un système canonique*

ORIENTATION

$$s \xleftrightarrow[u=v]{\sigma,p} t \Longrightarrow s \xrightarrow[S]{*} s' \xrightarrow[l\to r]{\sigma,p} t' \xleftrightarrow[AC]{*} t$$
$$\text{si } u \succ v \text{ et } l \to r \in \Psi(u,v)$$

DÉDUCTION

$$s \xleftarrow[l\to r]{\sigma,pA} t \xleftrightarrow[\equiv]{*} t' \xrightarrow[g\to d]{\sigma,pqB} u \Longrightarrow s \xleftrightarrow[S]{(\geq p)^*} s' \xleftrightarrow[u=v]{\sigma',p} u' \xleftrightarrow[S]{(\geq p)^*} u$$
$$\text{si } \begin{cases} \exists\theta \in \mathrm{CSU}_T(l|_q, g) \mid \sigma \equiv \theta\sigma' \\ u = v \in CP_T() \end{cases}$$

EFFACEMENT

$$s \xleftrightarrow[l=r]{\sigma,p} t \Longrightarrow s \xleftrightarrow[\equiv]{*} t$$
$$\text{si } l \equiv r$$

COMPOSITION

$$s \xrightarrow[l\to r]{\sigma,p} t \Longrightarrow s \xrightarrow[l\to r']{\sigma,p} u \xleftarrow[g\to d]{\theta\sigma,pq} t' \xleftrightarrow[\equiv]{*} t$$
$$\text{si } r \xrightarrow[g\to d]{\theta,q} r' (\text{et } s = s{\downarrow_S})$$

SIMPLIFICATION

$$s \xleftrightarrow[u=v]{\sigma,p} t \Longrightarrow s \xleftrightarrow[\equiv]{*} s' \xrightarrow[g\to d]{\theta\sigma,pq} t' \xleftrightarrow[u'=v]{\sigma,p} t$$
$$\text{si } u \xrightarrow[g\to d]{\theta,q} u' (\text{et } s = s{\downarrow_S})$$

RÉDUCTION

$$s \xrightarrow[l\to r]{\sigma,p} t \Longrightarrow s \xleftrightarrow[\equiv]{*} s' \xrightarrow[g\to d]{\theta\sigma,pq} t' \xleftrightarrow[l'=r]{\sigma,p} t$$
$$\text{si } \begin{cases} l \xrightarrow[g\to d]{\theta,q} l' \\ (l,r) \succ (g\theta, d\theta) \end{cases} \quad (\text{et } s = s{\downarrow_S})$$

Figure 4.8 : Réécriture de preuve : règles reflétant les règles d'inférence

règle $P \Longrightarrow Q$	$C(P)$	$C(Q)$	preuve $P \succ_P Q$
NORMALISATION DES SUBSTITUTIONS	$\langle\{s\downarrow_p, t\downarrow_p\}, \{s,t\}, \ldots\rangle$, $\langle\perp, \ldots\rangle$	$\langle\{s'\downarrow_p, t'\downarrow_p\}, \{s',t'\}, \ldots\rangle$, $\langle\perp, \ldots\rangle$	même première composante et $s \succ s'$ et $t \succ t'$
NORMALISATION DES ÉQUATIONS	$\langle\{s\downarrow_p, t\downarrow_p\}, \{s,t\}, \ldots\rangle$, $\langle\perp, \ldots\rangle$	$\langle\{s'\downarrow_p, t'\downarrow_p\}, \{s',t'\}, \ldots\rangle$, $\langle\perp, \ldots\rangle$	même première composante et $s \succ s'$ et $t \succ t'$
NORMALISATION À UNE POSITION PARALLÈLE	$\langle\{s\downarrow_p, t\downarrow_p\}, \{s,t\}, \ldots\rangle$, $\langle\perp, \ldots\rangle$	$\langle\{s'\downarrow_p, t'\downarrow_p\}, \{s',t'\}, \ldots\rangle$, $\langle\perp, \ldots\rangle$	même première composante et $s \succ s'$ et $t \succ t'$
NORMALISATION À UNE POSITION ANCÊTRE	$\langle\{s\downarrow_p, t\downarrow_p\}, \{s,t\}, \ldots\rangle$, $(\langle\perp, \ldots\rangle)^*$	$(\langle\{s'\downarrow_p, t'\downarrow_p\}, \{s',t'\}, \ldots\rangle)^*$, $(\langle\perp, \ldots\rangle)^*$	même première composante et $s \succ s'$ et $s \succ t$ (car $u \succ v$)

Figure 4.9 : Terminaison de la réécriture des preuves (règles de normalisation)

règle $P \Longrightarrow Q$	$C(P)$	$C(Q)$	preuve $P \succ_P Q$
S-CONFLUENCE	$\langle\perp, \{t\}, \ldots\rangle$, $\langle\perp, \{t'\}, \ldots\rangle$	$\langle\perp, \{s\}, \ldots\rangle$, $\langle\perp, \{u\}, \ldots\rangle$	$t \succ s$ et $t' \succ u$
COMMUTATION DES PICS EN PARALLÈLE	$\langle\{t\}, \ldots\rangle$, $\langle\{t\}, \ldots\rangle$	$\langle\{s\downarrow_q, v\downarrow_q\}, \ldots\rangle$, $\langle\{v\downarrow_p, u\downarrow_p\}, \ldots\rangle$	$t = t\downarrow_S$, $t \succ s \succeq s\downarrow_p$, $t \succ v$ et $t \succ u$
COMMUTATION DES PICS ANCÊTRES	$\langle\{t\}, \ldots\rangle$, $\langle\{t\}, \ldots\rangle$	$(\langle\{\ldots\}, \ldots\rangle)^*$, $(\langle\{\ldots\}, \ldots\rangle)^*$	analogue au cas ci-dessus

Figure 4.10 : Terminaison de la réécriture des preuves (règles de commutation des pics)

règle $P \Longrightarrow Q$	$C(P)$	$C(Q)$	preuve $P \succ_P Q$
ORIENTATION			Par définition
INSTANCES CRITIQUES			Par définition
DÉDUCTION	$\langle\{t\},\ldots\rangle$, $\langle\perp,\ldots\rangle$, $\langle\{t'\},\ldots\rangle$	$(\langle\perp,\ldots\rangle)^*$, $\langle\{s'\downarrow_p, u'\downarrow_p\},\ldots\rangle$, $(\langle\perp,\ldots\rangle)^*$	$t = t\downarrow_S$, $t \succ s \succeq s'\downarrow_p$ et $t' \succ u \succeq u'\downarrow_p$
EFFACEMENT	$\langle\{s\downarrow_p, t\downarrow_p\},\ldots\rangle$	$\langle\perp,\ldots\rangle$	trivial
COMPOSITION	$\langle\{s\},\{s\}, l\sigma, r\sigma\rangle$	$\langle\{s\},\{s\}, l\sigma, r'\sigma\rangle$, $\langle c(t', pq, u), \{t'\},\ldots\rangle$, $\langle\perp,\ldots\rangle$	$r \succ r'$ et $s \succ t \equiv t' \succ u$
SIMPLIFICATION	$\langle\{s\downarrow_p, t\downarrow_p\},\ldots\rangle$	$\langle\perp,\ldots\rangle$, $\langle\{s'\},\ldots\rangle$, $\langle\{t'\downarrow_p, t\downarrow_p\},\ldots\rangle$	$s = s\downarrow_S$ et $s \equiv s' \succ t' \succeq t'\downarrow_p$
RÉDUCTION	$\langle\{s\},\{s\}, l\sigma, r\sigma\rangle$	$\langle\perp\rangle$, $\langle\{s'\},\{s'\}, g\theta\sigma, d\theta\sigma\rangle$, $\langle\{t'\downarrow_p, t\downarrow_p\},\ldots\rangle$	$s = s\downarrow_S$, $(l,r) \succ\!\!\succ (g\theta, d\theta)$, $s \equiv s' \succ t'$ et $s \succ t$

Figure 4.11 : Terminaison de la réécriture des preuves (règles reflétant les inférences)

modulo AC. Si une règle de S est R/AC-réductible, il suffit d'inter-réduire $R \cup S$ pour obtenir un système canonique modulo AC.

Démonstration. $R \cup S$ est convergent modulo AC, et il est inter-réduit sous l'hypothèse qu'aucune règle de S n'est R/AC-réductible. □

Nous verrons dans la section suivante un exemple illustrant ce cas pathologique où $R \cup S$ n'est pas inter-réduit.

Une remarque importante à donner est qu'il n'existe pas de théorie équationnelle qui aurait un système canonique S-normalisé mais pas de système AC-canonique, c'est-à-dire que la notion de réécriture S-normalisée n'apporte rien du point de vue de la puissance d'expression.

Le résultat d'unicité qui suit n'est donc pas surprenant.

Théorème 4.10 *Soit $\succ$ un ordre de simplification compatible avec AC et soient R_1 et R_2 deux systèmes canoniques pour la réécriture S-normalisée, définissant la même théorie équationnelle, et qui soient tous deux inclus dans $\succ$. Alors R_1 et R_2 sont égaux modulo AC à renommage près.*

Démonstration. La preuve est évidemment très semblable au cas AC : si $l \to r$ est une règle de R_1 on trouve facilement une règle de $R_2 \cup S$ qui réduit l_1, mais ce ne peut pas être une règle de S car R_1 est inter-réduit, et on conclut de la même façon que dans le cas AC. □

4.4 Modularité

Intéressons-nous maintenant au mélange de plusieurs systèmes canoniques. Par exemple, que si passe-t-il si l'on veut compléter de façon normalisée par un S qui contient la théorie des groupes abéliens pour $+$, $-$ et 0; et par exemple l'idempotence pour un opérateur AC $*$ différent de $+$? Une chose évidente qu'il faut remarquer d'abord, c'est que l'on ne pourra rien dire si l'on mélange deux systèmes S_1 et S_2 qui partagent des symboles de fonctions : la confluence de $S_1 \cup S_2$ n'est pas assurée.

Lemme 4.11 *Si S_1 et S_2 sont sur des signatures disjointes, et tels que $S_1 \subseteq \succ$ et $S_2 \subseteq \succ$ pour un même ordre de réduction $\succ$, alors $S_1 \cup S_2$ est convergent.*

Démonstration. $S_1 \cup S_2 \subseteq \succ$ donc $S_1 \cup S_2$ termine, et ses paires critiques sont $CP(S_1 \cup S_2) = CP(S_1) \cup CP(S_2)$ car comme les signatures sont disjointes, il n'y a aucune superposition possible entre une règle de S_1 et une règle de S_2, et donc les paires critiques sont toutes joignables. □

On peut remarquer qu'ici ce résultat est élémentaire, on n'a pas besoin d'utiliser le résultat beaucoup plus difficile de Toyama [86] montrant que la confluence est modulaire pour des systèmes disjoints, même en l'absence de la terminaison (et de toute façon ce résultat s'applique pour la réécriture standard, pas la réécriture modulo AC).

Grâce à ce lemme, cela a un sens de parler de réécriture normalisée et de complétion modulo $S_1 \cup S_2$.

Théorème 4.12 *On suppose toujours que S_1 et S_2 sont sur des signatures disjointes et inclus dans le même ordre de réduction.*

(i) Si le couple (Θ_1, Ψ_1) est S_1-normalisant et le couple (Θ_2, Ψ_2) est S_2-normalisant, alors le couple $(\Theta_1 \cup \Theta_2, \Psi_1 \cup \Psi_2)$ est $S_1 \cup S_2$-normalisant.

(ii) Si pour un ensemble de règles R, R/S_1 et R/S_2 sont convergents, alors $R/(S_1 \cup S_2)$ est convergent.

Démonstration. Immédiate. □

La conséquence est que pour compléter de façon normalisée par $S_1 \cup S_2$, il suffit de considérer la réunion des couples critiques pour S_1 et pour S_2. Ceci est utilisé dans notre implantation, où l'on peut travailler modulo un ensemble de théories prédéfinies (voir annexe A).

4.5 Couples normalisants pour des théories particulières

Nous allons voir qu'il est intéressant d'avoir des couples normalisants différents de la forme générale pour certaines théories particulières. Nous allons voir que l'algorithme de calcul des bases standard est exactement une instance de la complétion normalisée modulo les anneaux commutatifs, pour un certain couple normalisant.

Dans certain cas, on n'a pas de formule meilleure que la formule générale pour un couple normalisant. Néanmoins, même dans ce cas-là il y a une « optimisation » intéressante : dans le cas où les règles de S sont linéaires gauches. en effet, dans ce cas il est possible d'expliciter la formule générale de manière à se passer complètement de l'unification AC. Ce sera le cas pour AC1, A1. Par contre, dans les autres cas, par exemple ACI ($x + x \to x$ non linéaire gauche) ou les groupes abéliens ($x + (-x) \to 0$ non linéaire gauche), on ne peut pas se passer complètement de l'unification AC, seulement en partie, mais c'est déjà intéressant.

4.5.1 AC1(+,0)

Dans le cas AC1, il n'y a pas de couple normalisant meilleur que celui donné par la formule générale, néanmoins on peut quand même l'expliciter, de manière à le calculer directement sans passer par l'unification AC.

Définition 4.13 *Posons*

$$\begin{cases} \Theta_{\mathrm{AC1}}(u,v) = \{u\theta = v\theta \mid \theta = x \mapsto 0, u \trianglerighteq x + w\} \\ \Psi_{\mathrm{AC1}}(u,v) = \{u \to v\} \cup \{x + v \to x \ si \ u = 0\} \end{cases}.$$

Proposition 4.14 *Le couple* $(\Theta_{\mathrm{AC1}}, \Psi_{\mathrm{AC1}})$ *est* AC1*-normalisant.*

Démonstration. Il s'agit simplement de voir ce que la définition générale donne dans le cas AC1 : il faut unifier $x + 0$ avec les sous-termes de u, les seules solutions consistent à instancier une variable de u sous un + par 0; sauf dans le cas précis où u est lui-même 0, auquel cas il y aussi une solution qui donne l'équation critique $x + v = x$ où x est une « nouvelle » variable (on peut noter de toute façon que v est nécessairement clos donc x peut être n'importe quelle variable). □

On peut remarquer que ce dernier cas particulier ne peut pas arriver si 0 est minimal, ce qui sera le cas dans tous les exemples que nous donnerons.

Exemple. Supposons que l'on complète la théorie des groupes abéliens modulo AC1. La figure 4.12 donne un script de la complétion telle qu'elle est obtenue dans notre implantation.

On peut remarquer en particulier que l'on a obtenu l'équation $-0 = 0$ sans calculer la moindre paire critique, donc sans appeler l'unification AC. Ceci améliore donc l'efficacité de la complétion.

La figure 4.13 donne les résultats obtenus pour la complétion des groupes abéliens modulo AC puis AC1. On constate un (léger) gain en efficacité en passant à AC1 : le nombre de superpositions et le nombre de paires critiques obtenues est moindre, il en résulte un plus petit nombre de réductions (indiqué par le nombre d'appels au filtrage). le temps de calcul n'est que très légèrement inférieur, probablement à cause du temps passé à calculer les instances critiques.

Il faut remarquer que la complétion modulo AC1 effectuée ici utilise l'unification AC plutôt que AC1 car nous n'avons pas encore d'implantation de l'unification AC1. Il est déjà intéressant de constater que la complétion AC1 est plus efficace que l'unification AC même sans utiliser l'unification AC1. On peut sans doute espérer un gain plus sensible grâce à l'unification AC1, qui doit en théorie réduire le nombre de paires critiques.

<table>
<tr><th>Source</th><th>Script de la complétion</th></tr>
<tr><td><pre>
operators
  + : AC
  0 : constant
  - : unary
  x,y : variable

theory AC1(+,0)

axioms
  x+-(x) = 0;

order
  rpo ->+>0 ; + mul

end
</pre></td><td><pre>
Orientation de l'equation x+-(x) = 0
en
{x+-(x) -> 0} (1 regles)
Avec les equations critiques :
{0+-(0) = 0} (1 equations)
Orientation de l'equation -(0) = 0
en
{-(0) -> 0} (1 regles)
Avec les equations critiques :
{}
Orientation de l'equation x = -(-(x))
en
{-(-(x)) -> x} (1 regles)
Avec les equations critiques :
{}
Orientation de l'equation y+-(x+y) = -(x)
en
{y+-(x+y) -> -(x)} (1 regles)
Avec les equations critiques :
{y+-(0+y) = -(0),
0+-(x+0) = -(x)} (2 equations)
Orientation de l'equation -(x+y) = -(y)+-(x)
en
{-(x+y) -> -(y)+-(x)} (1 regles)
Avec les equations critiques :
{-(0+y) = -(y)+-(0),
-(x+0) = -(0)+-(x)} (2 equations)

Resultat :
{x+-(x) -> 0,
-(-(x)) -> x,
-(0) -> 0,
-(x+y) -> -(y)+-(x)} (4 regles)

Nombre d'appels au filtrage AC   : 10422
Nombre d'appels reussis   : 756 (7%)
Nombre d'appels a l'unification AC : 30
Nombre de solutions trouvees   : 215 (7 en moyenne)

Temps utilisateur : 22.330 s
Temps systeme : 0.570 s
</pre></td></tr>
</table>

Figure 4.12 : Complétion des groupes abéliens modulo $AC1(.,1)$

4.5.2 ACI, AC1I, AC0 et ACN

Les quatre théories en question sont présentées par les systèmes canoniques donnés dans le tableau de la figure 4.14.

Pour l'ordre $\succ$ il n'y a quasiment pas de contrainte car toutes ces règles s'orientent grâce à la propriété sous-terme, sauf la dernière qui nécessite que 0 soit minimal.

Pour ce qui est des couples normalisants, les axiomes non linéaires d'idempotence et de nilpotence nécessitent d'appeler l'unification AC, mais pour les autres on a une formule très simple.

	Groupes abéliens, modulo AC	Groupes abéliens, modulo AC1
Équations données	AC(+) $x + 0 = x$ $x + -(x) = 0$	AC1(+, 0) $x + -(x) = 0$
Règles obtenues	$x + 0 \rightarrow x$ $x + -(x) \rightarrow 0$ $-(0) \rightarrow 0$ $-(-(x)) \rightarrow x$ $-(x + y) \rightarrow -(x) + -(y)$	$x + -(x) \rightarrow 0$ $-(0) \rightarrow 0$ $-(-(x)) \rightarrow x$ $-(x + y) \rightarrow -(x) + -(y)$
Temps de calcul	20"94	19"56
nombre d'appels à l'unification AC	39	30
nombre de paires critiques	242	215
nombre d'appels au filtrage AC	9517	8917

Figure 4.13 : Résultats pratiques: groupes et groupes abéliens

Théorie S	système canonique	$\Theta_S(u, v)$
$ACI(+)$	$x + x \rightarrow x$	$\{u\sigma = v\sigma \mid \sigma = mgu_{AC}(l_1, l_2), u \trianglerighteq l1 + l2\}$
$AC1I(+, 0)$	$x + x \rightarrow x$, $x + 0 \rightarrow x$	$\Theta_{AC1}(u, v) \cup \Theta_{ACI}(u, v)$
$AC0(., 0)$	$x.0 \rightarrow 0$	$\{u\theta = v\theta \mid \theta = x \mapsto 0, u \trianglerighteq x.w\}$
$ACN(+, 0)$	$x + x \rightarrow 0$	$\{u\sigma = v\sigma \mid \sigma = mgu_{AC}(l_1, l_2), u \trianglerighteq l1 + l2\}$

Figure 4.14 : Ensemble Θ_S pour S =ACI, AC1I, AC0 et ACN

D'autre part on pose $\Psi_S(u, v) = \{u \rightarrow v\}$ pour tous les S précédents.

Proposition 4.15 *Pour S = ACI, $AC1I$, $AC0$, ACN, le couple défini dans le tableau précédent est S-normalisant.*

Enfin, concernant la théorie T modulo laquelle on unifie dans le calcul des paires critiques, nous pouvons utiliser l'unification ACI ou AC1I pour son efficacité, comme nous l'avons déjà remarqué.

Exemple. Complétion de la théorie des anneaux booléens modulo $ACN(+, 0)$ et AC1I(.,1). Le script est donné figure 4.15. 0 et 1 représentent respectivement les constantes *faux* et *vrai*, + est le *ou exclusif*, . est le *et*, − est la négation, tandis que i représente l'inverse pour + (son utilité apparaîtra lorsque l'on complétera modulo les groupes abéliens.

<table>
<tr><th>Source</th><th>Script de la complétion</th></tr>
<tr><td><pre>
operators
  +,. : AC
  0,1 : constant
  i,-   : 1
  x,y,z : variable

theory AC1N(+,0) AC1I(.,1)

axioms
  x+i(x) = 0;
  x.(y+z) = (x.y)+(x.z);
  x+-(x) = 1;

order rpo ->1>.>i>+>0

end
</pre></td><td><pre>
Orientation de l'equation x+-(x) = 1
en
{x+-(x) -> 1} (1 regles)
Avec les equations critiques :
{...} (3 equations)
Orientation de l'equation 1 = -(0)
en
{-(0) -> 1} (1 regles)
Orientation de l'equation x+i(x) = 0
en
{x+i(x) -> 0} (1 regles)
Avec les equations critiques :
{...} (3 equations)
Orientation de l'equation 0 = i(0)
en
{i(0) -> 0} (1 regles)
Orientation de l'equation x = i(i(x))
en
{i(i(x)) -> x} (1 regles)
Orientation de l'equation x+1 = -(i(x))
en
{-(i(x)) -> x+1} (1 regles)
Orientation de l'equation -(x) = i(x)+1
en
{-(x) -> i(x)+1} (1 regles)
Orientation de l'equation x = i(i(y+x)+y)
en
{i(i(y+x)+y) -> x} (1 regles)
Avec les equations critiques :
{...} (28 equations)
Orientation de l'equation i(x) = x
en
{i(x) -> x} (1 regles)
Orientation de l'equation x.(y+z) = (x.y)+(x.z)
en
{x.(y+z) -> (x.y)+(x.z)} (1 regles)
Avec les equations critiques :
{...} (33 equations)
Orientation de l'equation x.0 = 0
en
{x.0 -> 0} (1 regles)
Avec les equations critiques :
{...} (7 equations)

Resultat :
{i(x) -> x,
-(x) -> x+1,
x.0 -> 0,
x.(y+z) -> (x.y)+(x.z)} (4 regles)

Nombre d'appels au filtrage AC   : 15221
Nombre d'appels reussis   : 961 (6%)
Nombre d'appels a l'unification AC : 43
Nombre de solutions trouvees   : 202 (4.70 en moyenne)
Temps utilisateur : 19.220 s
Temps systeme : 0.430 s
</pre></td></tr>
</table>

Figure 4.15 : Complétion des anneaux booléens modulo $ACN(+,0)$ et $AC1I(.,1)$

4.5.3 Groupes abéliens

Pour S nous avons le système de règles convergent modulo AC bien connu :

$$AG \left\{ \begin{array}{l} x+0 \to x \\ x+(-x) \to 0 \\ -0 \to 0 \\ -(-x) \to x \\ -(x+y) \to (-x)+(-y) \end{array} \right. .$$

Comme ordre $\succ$ nous prenons le *rpo* avec une précédence de la forme $\mathcal{F} > - > + > 0$, et tel que $+$ a le statut multi-ensemble

Nous donnons maintenant un couple AG-normalisant « optimisé », qui suppose que l'ordre $\succ$ est un *rpo* avec une précédence telle que $- > + > 0$ et aucun autre symbole n'est plus petit que $-$, et enfin tel que $+$ a le statut multi-ensemble. Il nous faut introduire une notion préliminaire de *symétrisation* inspirée de celle qu'a définie Le Chenadec [62].

Nous utiliserons la notation nt pour le terme $\underbrace{t+\cdots+t}_{n \text{ fois}}$.

Définition 4.16 *Le* symétrisé *d'une paire de termes $\{u,v\}$, que l'on note $sym(u,v)$, est obtenu de la façon suivante : si $u = n_1u_1 + \cdots + n_ku_k$ et $v = m_1t_1 + \cdots + m_lt_l$, avec $\forall j \geq 2 u_1 \succ u_j$ et $\forall j u_1 \succ t_j$, alors*

$$sym(u,v) = (n_1, u_1, -n_2u_2 - \cdots - n_ku_k + m_1t_1 + \cdots + m_lt_l)$$

Remarquons que s'il n'y a pas de termes u_i ou v_j maximum, le symétrisé n'est pas défini.

Définition 4.17 *Pour un couple (u,v) arbitraire possédant un symétrisé (n,s,t), posons*

$$\begin{array}{rcl} \Psi_{AG}(u,v) & = & \left\{ \begin{array}{l} ns \to t \\ -s \to (n-1)s + (-t) \ si\ n \geq 2 \end{array} \right. \\ \Theta_{AG}(u,v) & = & \Theta_{AC1}(ns,t) \cup \Sigma_1(ns,t) \cup \Sigma_2(ns,t) \end{array}$$

où

$$\begin{array}{rcl} \Sigma_1(u,v) & = & CP_{AC}(u \to v, x+(-x) \to 0) \\ \Sigma_2(u,v) & = & \{u\sigma = v\sigma | \sigma = x \mapsto 0 \ ou\ -y \ ou\ y+z \ si\ u \trianglerighteq -x\} \end{array}$$

Si (u,v) ne possède pas de symétrisé, c'est-à-dire s'il n'y pas de termes maximum dans u et v, l'équation $u=v$ sera considérée comme inorientable.

Ce couple AG-normalisant généralise au cas non clos la méthode d'orientation par symétrisation de Le Chenadec [62].

Proposition 4.18 *Le couple (Θ_{AG}, Ψ_{AG}) est AG-normalisant.*

Démonstration. Puisque l'on travaille modulo AG, l'équation $u=v$ est équivalente à l'équation $ns=t$, sur laquelle on peut alors calculer la formule générale de couple normalisant. La règle supplémentaire est due à un cas critique modulo l'inverse, que l'on a orienté immédiatement en règle, ce qui est complet car elle ne donne pas de cas critiques en plus de ceux de l'équation de départ. □

<table>
<tr><th>Source</th><th>Script de la complétion</th></tr>
<tr><td><pre>operators
  +,. : AC
  0,1 : constant
  -   : 1
  x,y,z,t,u : variable

theory AG(+,0,-) AC1(.,1)

axioms
  x.(y+z) = (x.y)+(x.z);

order
  rpo 1>.>->+>0 ;
  + mul, . mul

end</pre></td><td><pre>Orientation de l'equation x.(y+z) = (x.y)+(x.z)
en
{x.(y+z) -> (x.y)+(x.z)} (1 regles)
Avec les equations critiques :
{1.(y+z) = (1.y)+(1.z),
x.(0+z) = (x.0)+(x.z),
x.(y+0) = (x.y)+(x.0),
x.0 = (x.y)+(x.(z+-(y+z))),
x.0 = (x.y)+(x.-(y)),
x.0 = (x.-(y))+(x.y),
x.0 = (x.(y+-(y+z)))+(x.z),
x.t = (x.y)+(x.(z+-(y+z)+t)),
x.z = (x.y)+(x.(-(y)+z)),
x.z = (x.(y+-(y)))+(x.z),
x.z = (x.-(y))+(x.(y+z)),
x.t = (x.(y+-(y+z)))+(x.(z+t)),
x.(z+t) = (x.(y+-(y)+z))+(x.t),
x.(z+t) = (x.(-(y)+z))+(x.(y+t)),
x.(t+u) = (x.(y+-(y+z)+t))+(x.(z+u)),
x.(y+t) = (x.y)+(x.(z+-(z)+t)),
x.(z+u) = (x.(y+z))+(x.(t+-(y+t)+u)),
x.(z+t) = (x.(y+z))+(x.(-(y)+t)),
x.z = (x.(-(y)+z))+(x.y),
x.t = (x.(y+-(y+z)+t))+(x.z),
x.y = (x.y)+(x.(z+-(z))),
x.z = (x.(y+z))+(x.(t+-(y+t))),
x.z = (x.(y+z))+(x.-(y))} (23 equations)
Orientation de l'equation x.y = (x.y)+(x.0)
en
{x.0 -> 0} (1 regles)
Avec les equations critiques :
{(x.1)+(x.0) = x.1,
(1.y)+(1.0) = 1.y} (2 equations)
Orientation de l'equation 0 = (x.-(z))+(x.z)
en
{x.-(z) -> -(x.z)} (1 regles)
Avec les equations critiques :
{(1.-(z))+(1.z) = 0,
(x.-(1))+(x.1) = 0} (2 equations)

Resultat :
{x.-(z) -> -(x.z),
x.0 -> 0,
x.(y+z) -> (x.y)+(x.z)} (3 regles)

Nombre d'appels au filtrage AC   : 2678
Nombre d'appels reussis   : 174 (6%)
Nombre d'appels a l'unification AC : 8
Nombre de solutions trouvees   : 39 (4.88 en moyenne)
Temps utilisateur : 6.490 s
Temps systeme : 0.180 s</pre></td></tr>
</table>

Figure 4.16 : Théorie des anneaux commutatifs modulo $AG(+,0,-)$ et $AC1(.,1)$

	modulo AC	modulo AC1	modulo AG	modulo AG∪AC1
Temps de calcul	58”15	50”16	6”05	6”49
nombre d'appels à l'unification AC	136	76	12	8
nombre de paires critiques	537	412	46	39
nombre d'appels au filtrage AC	23925	21272	2562	2678

Figure 4.17 : Résultats pratiques : anneaux commutatifs modulo AC, AC1 et AG

Exemple. Complétion de la théorie des anneaux commutatifs. La figure 4.16 montre le script de la complétion modulo $AG(+,-,0)$ et $AC1(.,1)$. Sur cet exemple, *toutes* les nouvelles règles ont été trouvées comme instances critiques de la règle de distributivité.

Cet exemple suscite deux remarques :

1. l'orientation de l'équation $x.y = (x.y) + (x.0)$ donne la règle $x.0 \to 0$ et les équations critiques $(x.1) + (x.0) = x.1$ et $(1.y) + (1.0) = 1.y$. On aurait eu uniquement $1.0 = 0$ comme équation critiques si l'on avait utilisé un couple normalisant optimal pour la théorie $AG(+,0,-) \cup AC1(.,1)$, meilleur que la réunion des couples normalisants de chacune des deux parties disjointes;

2. le nombre d'équations critiques de $x.(y+z) = (x.y) + (x.z)$ (23) est trop grand par rapport à ce que donne la théorie, parce que dans notre implantation on ne teste pas si la substitution est S-irréductible. Sinon on aurait obtenu seulement les onze équations :

$$\left\{\begin{array}{l}
1.(y+z) = (1.y) + (1.z) \\
x.(0+z) = (x.0) + (x.z) \\
x.(y+0) = (x.y) + (x.0) \\
x.0 = (x.y) + (x.-(y)) \\
x.0 = (x.-(y)) + (x.y) \\
x.z = (x.y) + (x.(-(y)+z)) \\
x.z = (x.-(y)) + (x.(y+z)) \\
x.(z+t) = (x.(-(y)+z)) + (x.(y+t)) \\
x.(z+t) = (x.(y+z)) + (x.(-(y)+t)) \\
x.z = (x.(-(y)+z)) + (x.y) \\
x.z = (x.(y+z)) + (x.-(y))
\end{array}\right.$$

La figure 4.17 compare avec la complétion des anneaux commutatifs modulo AC, AC1 et AG. L'efficacité de la complétion modulo AG est indiscutable.

Exemple. La théorie des anneaux booléens modulo AG(+,0,i) et AC1I(.,1). Voir figure 4.18.

Il est assez amusant de constater que la règle de nilpotence a été déduite de l'équation $i(x) + i(x) = 0$! Mais cela provient probablement d'un mauvais choix dans la stratégie de choix de la prochaine équation à orienter...

Source	Script de la complétion
operators	Orientation de l'equation x+-(x) = 1
+,. : AC	en
0,1 : constant	{-(x) -> 1+i(x)} (1 regles)
i,- : 1	Avec les equations critiques :
x,y,z : variable	{}
	Orientation de l'equation x.(y+z) = (x.y)+(x.z)
theory AG(+,0,i) AC1I(.,1)	en
	{x.(y+z) -> (x.y)+(x.z)} (1 regles)
axioms	Avec les equations critiques :
x.(y+z) = (x.y)+(x.z);	{...} (29 equations)
x+-(x) = 1;	Orientation de l'equation x.y = (x.y)+(x.0)
	en
order rpo ->1>.>->+>0	{x.0 -> 0} (1 regles)
	Avec les equations critiques :
end	{...} (32 equations)
	Orientation de l'equation 0 = (x.i(z))+(x.z)
	en
	{x.i(z) -> i(x.z)} (1 regles)
	Avec les equations critiques :
	{...} (32 equations)
	Orientation de l'equation i(x)+i(x) = 0
	en
	{x+x -> 0,
	i(x) -> i(0)+x} (2 regles)
	Avec les equations critiques :
	{...} (7 equations)
	Resultat :
	{i(x) -> x,
	-(x) -> 1+x,
	x.0 -> 0,
	x+x -> 0,
	x.(y+z) -> (x.y)+(x.z)} (5 regles)
	Nombre d'appels au filtrage AC : 8835
	Nombre d'appels reussis : 636 (7%)
	Nombre d'appels a l'unification AC : 24
	Nombre de solutions trouvees : 139 (5.79 en moyenne)
	Temps utilisateur : 11.390 s
	Temps systeme : 0.260 s

Figure 4.18 : Théorie des anneaux booléens modulo $AG(+,0,-)$ et $AC1I(.,1)$

On peut également remarquer que dans cet exemple on se trouve dans le cas particulier du théorème d'unicité : $R \cup S$ n'est pas inter-réduit ! En effet la règle $x + i(x) \rightarrow 0$ de S est réductible par $i(x) \rightarrow x$.

Nous verrons d'autres exemples de complétion modulo AG, dans le cas clos, dans le chapitre suivant.

4.5.4 Anneaux commutatifs

Pour S nous avons le système de règles convergent modulo AC classique :

$$CR \left\{ \begin{array}{l} x + 0 \to x \\ x + (-x) \to 0 \\ -0 \to 0 \\ -(-x) \to x \\ -(x+y) \to (-x) + (-y) \\ x * (y+z) \to (x*y) + (x*z) \\ x * 1 \to x \\ x * 0 \to 0 \\ x * (-y) \to -(x*y) \end{array} \right. .$$

Comme ordre $\succ$ nous prenons le *apo* avec une précédence de la forme $\mathcal{F} > * > 1 > - > + > 0$, et tel que $+$ et $*$ ont le statut multi-ensemble (voir paragraphe 2.4.3)

Pour orienter une équation $u = v$ normalisée nous utilisons comme dans le cas des groupes une technique de symétrisation. Si le symétrisé n'est pas défini l'équation est considérée comme inorientable.

Définition 4.19 *Pour une équation $u = v$ posons $(n, s, t) = sym(u, v)$, et soit*

$$\Theta_{CR}(u,v) = \Theta_{AC1}(ns,t) \cup \Theta_{AC0}(ns,t) \cup \Sigma_1(ns,t) \cup \Sigma_2(ns,t) \cup \Sigma_3(ns,t)$$

avec

$$\left\{ \begin{array}{l} \Sigma_1(u,v) = CP_{AC}(u \to v, x + (-x) \to 0) \\ \Sigma_2(u,v) = \{u\sigma = v\sigma\} \textit{ pour tout } \sigma = x \mapsto 0 \textit{ ou } -y \textit{ ou } y+z \textit{ si } -x \in u \\ \Sigma_3(u,v) = \{u\sigma = v\sigma\} \textit{ pour tout } \sigma = x \mapsto y+z \textit{ ou } -y \textit{ si } x*w \in u \end{array} \right.$$

et

$$\Psi_{CR}(u,v) = \{ns \to t\}$$

augmenté des trois règles suivantes si $n > 1$:

$$\left\{ \begin{array}{l} -s \to (n-1)s + (-t) \\ nx * s \to x * t \\ -x * s \to (n-1)x * s + (-x * t) \end{array} \right.$$

où x est une nouvelle variable.

Proposition 4.20 *Le couple (Θ_{CR}, Ψ_{CR}) est CR-normalisant.*

Démonstration. Puisque l'on travaille modulo CR, comme dans le cas AG on sait que l'équation $u = v$ est équivalente à l'équation $ns = t$, sur laquelle on peut alors calculer la formule générale de couple normalisant. Les trois règles supplémentaires sont dues à des cas critiques modulo la distributivité et l'inverse, de façon analogue au cas des groupes. □

<table>
<tr><th>Source</th><th>Script de la complétion</th></tr>
<tr><td>

```
operators
  +,. : AC
  &,* : AC
  0,1 : constant
  z,un : constant
  -   : 1
  O   : 1
  x,y,z : variable
  H : unary

theory  CR(+,0,-,.,1)
        CR(&,zero,O,*,un)

axioms
  H(x+y) = H(x) & H(y);
  H(x.y) = H(x) * H(y);

order
  rpo H>1>.>->+>0,
      H>un>*>O>&>zero

end
```

</td><td>

```
Orientation de l'equation H(x.y) = H(x)*H(y)
en
{H(x.y) -> H(x)*H(y),
H(x.y) -> H(x)*H(y)} (2 regles)
Avec les equations critiques :
{H(1.y) = H(1)*H(y),
H(x.1) = H(x)*H(1)} (2 equations)
Orientation de l'equation H(x) = H(x)*H(1)
en
{H(x)*H(1) -> H(x),
H(x)*H(1) -> H(x)} (2 regles)
Avec les equations critiques :
{}
Orientation de l'equation H(x+y) = H(x) & H(y)
en
{H(x+y) -> H(x) & H(y),
H(x+y) -> H(x) & H(y)} (2 regles)
Avec les equations critiques :
{H(0+y) = H(0) & H(y),
H(x+0) = H(x) & H(0),
H(0) = H(x) & H(y+-(x+y)),
H(0) = H(x) & H(-(x)),
...} (22 equations)
Orientation de l'equation H(x) = H(x) & H(0)
en
{H(0) -> zero,
H(x) & H(0) -> H(x)} (2 regles)
Avec les equations critiques :
{}
Orientation de l'equation zero = H(-(y)) & H(y)
en
{H(-(y)) -> O(H(y)),
H(-(y)) & H(y) -> zero} (2 regles)
Avec les equations critiques :
{}

Resultat :
{H(0) -> zero,
H(x)*H(1) -> H(x),
H(x+y) -> H(x) & H(y),
H(x.y) -> H(x)*H(y),
H(-(y)) -> O(H(y))} (5 regles)

Nombre d'appels au filtrage AC     : 2918
Nombre d'appels reussis            : 171 (5%)
Nombre d'appels a l'unification AC : 23
Nombre de solutions trouvees       : 54 (2.35 en moyenne)
Temps utilisateur : 7.680 s
Temps systeme : 0.440 s
```

</td></tr>
</table>

Figure 4.19 : Théorie d'un « quasi » morphisme d'anneaux commutatifs modulo CR

Exemple. Complétion de la théorie d'un « quasi » morphisme d'anneaux commutatifs. Il s'agit d'un quasi morphisme d'anneau car on ne dit pas que l'image des neutres sont les neutres, ni que l'image de l'opposé est l'opposé de l'image, on dit seulement que l'image de la somme est la somme des images, ainsi que pour le produit. Voir figure 4.19.

On peut remarquer que l'on a montré que l'image de 0 était $zero$ et l'image de l'opposé est l'opposé de l'image, mais l'on n'a pas obtenu l'équation $H(1) = un$, qui n'est effectivement pas conséquence des équations données (si l'on pose $H(x) = zero$ pour tout x, on a un modèle).

Nous verrons d'autres exemples de complétion modulo CR, dans le cas clos, dans le chapitre suivant.

Chapitre 5

Décidabilité de quelques problèmes du mot E-clos

Dans ce chapitre, nous allons voir comment utiliser la complétion normalisée comme procédure de décision du problème du mot dans certaines classes de théories équationnelles. Nous allons montrer en fait que les classes des théories équationnelles E-closes pour un certain nombre de E ont un problème du mot uniformément décidable, par construction d'un système de réécriture E-normalisée convergent.

La méthode utilisée est en grande partie uniforme, et les résultats généraux, valables pour tout E sont présentés dans la première section. Les sections suivantes sont alors consacrées à l'étude de E particuliers, où à chaque fois l'on comble la partie « manquante » de la démonstration.

Nous avons ainsi par cette méthode une preuve de décidabilité du problème du mot E-clos valable aussi bien pour le cas où est réduit à AC, retrouvant ainsi le résultat de Narendran et Rusinowitch [71] et pour le cas où E définit la théorie des anneaux, retrouvant ainsi le calcul de bases standard sur $\mathbb{Z}$ (ou sur n'importe quel extension algébrique de dimension finie, entiers de Gauss par exemple) [19, 49], ainsi que le calcul des bases standard sur les corps finis; et nous obtenons également de nouveaux résultats, concernant par exemple l'idempotence, et de manière générale pour les cas où la signature contient des symboles libres non constants. Nous obtenons également comme cas particulier les différents algorithmes de complétion dans des algèbres finiment engendrées de Le Chenadec [62], car ce sont des complétions closes modulo les groupes, les groupes abéliens, etc.

5.1 Résultats généraux

Nous supposons dorénavant que l'ordre $\succ$ est total, nous avons vu comment faire pour satisfaire cette hypothèse dans la section 2.5. Nous supposons aussi que l'ensemble d'équations initial donné à compléter est clos, et que la complétion ne peut pas produire d'équation non close à partir d'équations closes, ou plutôt, hypothèse moins forte qui servira pour la complétion modulo les anneaux, que les équations produites sont toujours orientables.

Nous définissons d'abord la notion *d'ensemble générateur* d'un terme. Cette notion est une généralisation de la notion définie par Narendran et Rusinowitch [71]. L'idée est de se choisir un sous-ensemble de la signature qui contiendra tous les symboles apparaissant dans les règles de S. On montre ensuite que si l'on « oublie » dans un certain sens ces symboles, on obtient ce qu'on appelle l'ensemble générateur, et l'on montre que, sous une certaine hypothèse sur la stratégie, cet ensemble ne croît pas lors de la complétion, et cet ensemble, qui reste donc contenu dans un ensemble fini fixé, nous permettra ensuite de conclure à la terminaison.

Pour les définitions suivantes, on se fixe un sous-ensemble F de la signature $\mathcal{F}$.

Définition 5.1 *Soit u un terme aplati.* L'ensemble générateur *de u par rapport à F est défini par :*

$$\begin{cases} \gamma_F(u) = \{u\} \text{ si } \mathcal{H}\mathrm{ead}(u) \notin F \\ \gamma_F(u) = \bigcup_{1 \leq i \leq n} \gamma_F(u_i) \text{ si } u = f(u_1, \ldots, u_n) \text{ avec } f \in F \end{cases}$$

Intuitivement, $\gamma_F(u)$ est l'ensemble des sous-termes de u obtenus en effaçant les symboles de tête de u qui sont dans F. Il faut bien remarquer qu'un symbole de F qui apparaît sous un symbole qui n'est pas dans F n'est pas effacé. Par exemple :

$$\gamma_{+,-,0}(f(a, b + c) + (-d) + 0) = \{f(a, b + c), d\}$$

L'ensemble générateur de u doit être vu comme l'ensemble des termes (ne commençant pas par un symbole de F) qui permettent de construire u avec des symboles de F.

Définition 5.2 L'ensemble générateur *(par rapport à F) d'un couple de termes (u, v) est l'ensemble $\Gamma_F(u, v) = \gamma_F(u) \cup \gamma_F(v)$.*

Soit E un ensemble d'équations et R un ensemble de règles. L'ensemble générateur *de E et R est défini par :*

$$G_F(E, R) = \bigcup_{u=v \in E} \Gamma_F(u, v) \cup \bigcup_{u \to v \in R} \Gamma_F(u, v)$$

Nous allons montrer que pendant toute dérivation de la complétion à partir de E_0, l'ensemble générateur de E et R décroît (au sens large) pour l'ordre multi-ensemble $\succ_{mul}$, en fait pas toujours, mais seulement entre deux étapes où E et R ne sont plus simplifiables.

À partir de maintenant, nous supposons fixé un système de réécriture convergent S, et que F est exactement l'ensemble des symboles qui apparaissent dans les règles de S.

Lemme 5.3 *Soit un ensemble de règles arbitraire R contenant une règle $l \to r$ et deux termes u et v. On a*

$$u \underset{S}{\longrightarrow} v \overset{*}{\underset{S}{\longrightarrow}} v' \implies \gamma_F(v') \preceq_{mul} \gamma_F(u)$$
$$u \underset{l \to r}{\longrightarrow} v \overset{*}{\underset{R}{\longrightarrow}} v' \implies \gamma_F(v') \cup \Gamma_F(R) \preceq_{mul} \gamma_F(u) \cup \Gamma_F(R)$$

où v' n'est plus réductible.

Démonstration. Pour la première assertion, on a deux cas suivant que la position à laquelle on applique la règle est dans un sous-terme t de $\gamma_F(u)$ ou non :

1. si la réécriture a lieu dans le chapeau de tête (figure 5.1, 1$^{\text{er}}$cas), l'ensemble générateur reste inchangé;

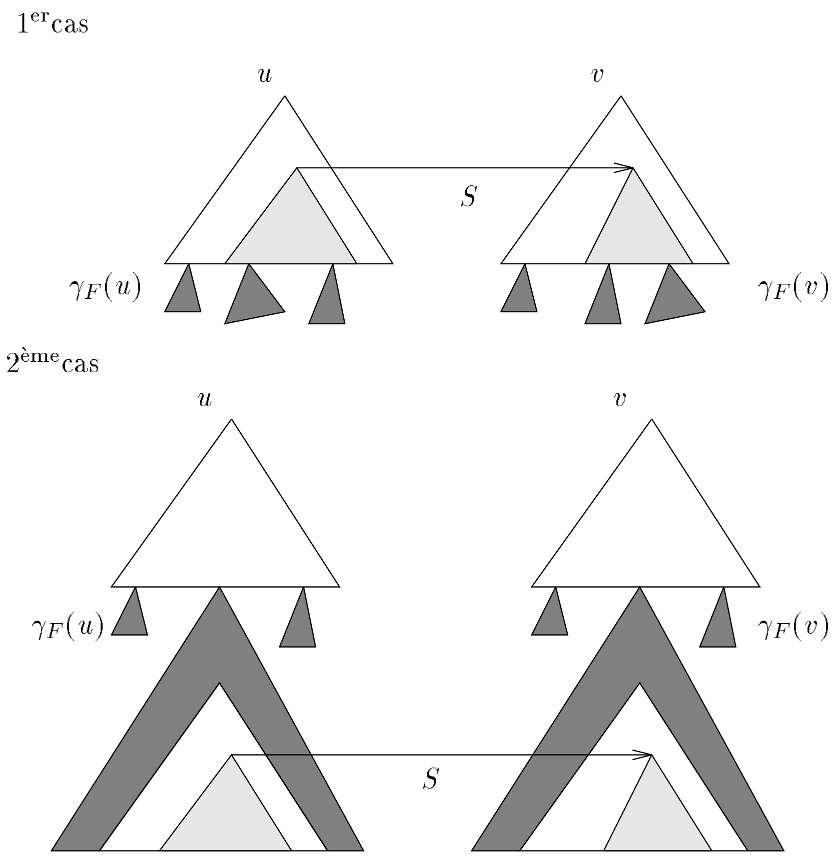

Figure 5.1 : Décroissance de l'ensemble générateur (1)

2. sinon, on réécrit dans un sous-terme générateur (figure 5.1, 2$^{\text{ème}}$cas), qui décroît, donc γ_F décroît, sauf s'il y a plusieurs occurrences du même sous-terme générateur, mais alors en réécrivant ces occurrences pour obtenir v', ces occurrences disparaissent, et $\gamma_F(v')$ est plus petit (voir exemple ci-dessous).

Pour la seconde assertion, il y a également deux cas :

1. s'il existe $t \in \gamma_F(u)$ tel que $u|_p$ est un sous-terme de t alors $\gamma_F(v) = \gamma_F(u) - \{t\} + \{t[r]_q\}$ (où q est la position de t tel que $t|_q = u|_p$). on conclut en remarquant que $t \succ t[r]_q$ (figure 5.2, 2$^{\text{ème}}$cas), avec la même remarque que précédemment : s'il y a plusieurs occurrences de t, il faut les réécrire aussi;

2. dans le cas contraire, l est formé de certains termes de $\gamma_F(u)$ et des symboles de F, en particulier $\gamma_F(l) \subseteq \gamma_F(u)$, et $\gamma_F(v)$ contient des termes qui sont déjà dans $\gamma_F(u)$ ou dans $\gamma_F(r)$ (figure 5.2, 1$^{\text{er}}$cas). □

Exemple. • Considérons l'étape de réécriture suivante :

$$a + b + c \xrightarrow[b+c \to d]{} a + d$$

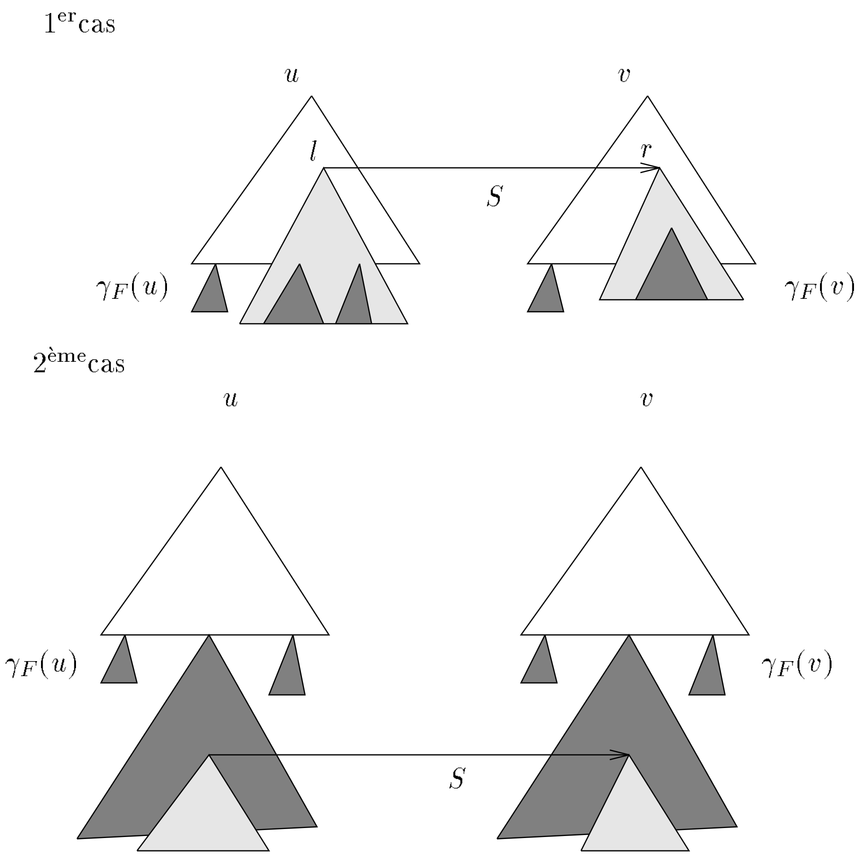

Figure 5.2 : Décroissance de l'ensemble générateur (2)

on a

$$\begin{cases} \gamma_F(u) \cup \gamma_F(l) \cup \gamma_F(r) = \{a, b, c\} \cup \{b, c\} \cup \{d\} = \{a, b, c, d\} \\ \gamma_F(v) \cup \gamma_F(l) \cup \gamma_F(r) = \{a, d\} \cup \{b, c\} \cup \{d\} = \{a, b, c, d\} \end{cases}$$

On voit sur cet exemple que si $\mathcal{F} - F$ ne contient que des constantes, alors G_F est simplement l'ensemble de ces constantes, et donc ne peut évidemment pas croître.

- autre exemple montrant pourquoi il faut regarder jusqu'à v' : si $R = \{a \to b, a \to c\}$

$$f(a) + f(a) \underset{a \to b}{\longrightarrow} f(a) + f(b) \underset{a \to c}{\longrightarrow} f(b) + f(c)$$

$\gamma_F(u) = \{f(a)\}$, $\gamma_F(v) = \{f(a), f(b)\}$ et $\gamma_F(v') = \{f(b), f(c)\}$. γ_F a crû localement, mais est finalement plus petit pour l'ordre multi-ensemble.

Théorème 5.4 *On suppose que le couple normalisant utilisé dans la complétion est tel que G_F reste constant (ou décroît) par les règles* Orientation *et* Déduction. *Soit E_0 un ensemble d'équations. Considérons une sous-séquence de dérivations*

$$E; R \overset{+}{\vdash} E'; R'$$

de la dérivation à partir de $E_0; \emptyset$*, telle que* $E'; R'$ *ne soient plus simplifiables, c'est-à-dire qu'aucune des règles de simplification* NORMALISATION*,* EFFACEMENT*,* SIMPLIFICATION*,* COMPOSITION *ou* RÉDUCTION *ne s'applique. Alors*

$$G_F(E, R) \succeq_{mul} G_F(E', R')$$

Démonstration. Immédiat en utilisant le lemme précédent, on a déjà remarqué que G_F pouvait augmenter « localement » mais se réduit à nouveau lorsque les simplifications sont terminées. □

Pour pouvoir appliquer ce théorème, il nous faut assurer que l'on simplifie en priorité. Ainsi, afin de continuer nos résultat généraux, nous avons maintenant besoin de mettre une hypothèse sur la stratégie.

Définition 5.5 *On dira que le contrôle* simplifie d'abord *si les règles de simplification* EFFACEMENT*,* NORMALISATION*,* SIMPLIFICATION*,* COMPOSITION *et* RÉDUCTION *sont prioritaires sur* ORIENTATION *et* DÉDUCTION*.*

Cette condition sur la stratégie est essentielle car sans elle il est possible que la complétion diverge bien que R_∞ reste fini. Ce phénomène peut avoir lieu déjà pour la complétion standard, mais personne ne l'avait encore remarqué à notre connaissance [66]. Voici un exemple :

Exemple. soient f et g deux opérateurs unaires. On peut engendrer à partir de l'ensemble d'équations $E_0 = \{ggx = gx, fgfx = fgx\}$ la dérivation équitable donnée figure 5.3.

On obtient un R_∞ fini : $R_\infty = \{ggx \to gx, fgfx \to fgx\}$. Cela se produit car on a utilisé une règle non simplifiée (3n+2) pour déduire une nouvelle équation.

Théorème 5.6 *Supposons que le contrôle simplifie d'abord. Alors si la complétion diverge,* R_∞ *est infini, et il contient une infinité de règles dont le symbole de tête du membre gauche est dans* F*.*

Démonstration. par l'absurde : si R_∞ est fini, il y a une étape j de la dérivation telle que $R_j = R_\infty$, et alors les étapes suivantes de la dérivation produisent des équations qui sont simplifiées puis effacées (elles sont prouvables par réécriture par R_∞) avant de pouvoir servir à déduire, et donc la dérivation doit être finie puisque $CP_T(R_\infty)$ est lui-même fini.

Maintenant supposons qu'il y ait une infinité de règles $l \to r$ telles que $\mathcal{H}\mathrm{ead}(l) \notin F$. La dérivation est de la forme

$$E_0; \emptyset \overset{*}{\vdash} E_{n_1}; R_{n_1} \overset{+}{\vdash} E_{n_2}; R_{n_2} \overset{+}{\vdash} \cdots$$

où tous les $E_{n_i}; R_{n_i}$ sont inter-réduits. la suite d'ensembles générateurs $G_F(E_{n_i}, R_{n_i})$ est donc décroissante (au sens large) pour $\succeq_{mul}$, elle est donc constante à partir d'un certain rang. Notons G_F^∞ cette constante. Les membres gauches des règles de R_∞ dont le symbole de tête n'est pas dans F sont dans G_F^∞, qui est fini, donc il y a des règles qui ont le même membre gauche. Mais ceci n'est pas possible si $\succ$ est total, car RÉDUCTION peut alors s'appliquer. □

Dans chacun des cas qui vont suivre, pour démontrer la terminaison de la complétion pour toute donnée, nous vérifions l'hypothèse que les règles ORIENTATION et DÉDUCTION font décroître $G_F(E, R)$, ce qui est toujours immédiat. Ensuite, si on suppose que la complétion ne termine pas, on sait qu'il y a une infinité de règles dont le membre gauche est constitué d'un « chapeau » formé de symboles de F et des termes dans G_F^∞.

	(1) $ggx = gx$
	(2) $fgfx = fgx$
Orientation de (1)	(3) $ggx \to gx$
Orientation de (2)	(4) $fgfx \to fgx$
Déduction entre (3) et (3)	
(superposition $gggx$)	(5) $ggx = ggx$
Effacement	(5) effacée
Déduction entre (4) et (4)	
(superposition $fgfgfx$)	(6) $fggfx = fgfgx$
Simplication de (6) par (4)	(7) $fggfx = fggx$
Orientation de (7)	(8) $fggfx \to fggx$
Déduction entre (8) et (4)	
(superposition $fggfgfx$)	(9) $fgggfx = fggfgx$
Simplification de (9) par (8)	(10)$fgggfx = fgggx$
Orientation de (10)	(11)$fgggfx \to fgggx$
Simplification de (8) avec (3) et (4)	
puis Effacement	(8) effacée
....	
	(3n+2) $fg^n fx \to fg^n x$
Déduction entre (3n+2) et (4)	
(superposition $fg^n fgfx$)	(3n+3) $fg^{n+1}fx = fg^n fgx$
Simplification de (3n+3) par (3n+2)	(3n+4)$fg^{n+1}fx = fg^{n+1}x$
Orientation de (3n+4)	(3n+5) $fg^{n+1}fx \to fg^{n+1}x$
Simplification de (3n+2) avec (3) et (4)	
puis Effacement	$(3n+2)$ effacée
....	

Figure 5.3 : Une dérivation infinie équitable engendrant un R_∞ fini

Remarque Les résultats de cette section sont valables quelle que soit la théorie S. Avant de voir comment l'on conclut la preuve de terminaison pour chacun des cas intéressants qui vont suivre, il est intéressant de voir un cas de non-terminaison, pour comprendre à quel endroit la démonstration ne fonctionne pas. Pour cela nous prenons pour S l'associativité, pour lequel il y a des théories closes avec problème du mot indécidable.

Prenons donc l'ensemble d'équations closes

$$E = \begin{cases} bab = abb \\ abbc = abc \end{cases}$$

On voit facilement que la complétion diverge en produisant l'ensemble infini de règles

$$R_\infty = \begin{cases} bab \to abb \\ \forall n \geq 2 \; ab^n c \to abc \end{cases}$$

Si l'on prend $F = \{.\}$, on a bien $G_F(R_\infty) = \{a, b, c\}$ fini, mais ce qui se produit en fait dans ce cas-là, c'est qu'il est possible de construire rien qu'à partir de cet ensemble fini une infinité de termes (les membres gauches des règles précédentes) deux à deux incomparables pour l'ordre sous-terme.

Dans chacun des cas qui vont suivre, nous utiliserons le lemme de Higman ou plutôt son corollaire 2.11 pour montrer qu'il n'est pas possible d'avoir une infinité de règles dont les membres gauches sont deux à deux incomparables pour l'ordre sous-terme.

5.2 Complétion close modulo AC1, ACI, AC1I, AC0 et ACN

Nous supposons qu'il y a dans la signature un nombre quelconque d'opérateurs AC, et que chacun d'eux vérifient l'une de ces cinq théories (ou éventuellement est seulement AC). Nous pouvons prendre comme ordre AC total l'ordre défini au chapitre 2.

5.2.1 Couples normalisants dans le cas clos

Si l'on regarde les couples normalisants du chapitre 4, on s'aperçoit que Θ devient vide dans le cas clos ! Nous pouvons donc utiliser comme couple normalisant, quelle que soit la théorie :

$$\begin{cases} \Theta_{\mathrm{AC1}I}(u,v) = \emptyset \\ \Psi_{\mathrm{AC1}I}(u,v) = \{u \to v\} \\ CP_T = CP_{\mathrm{AC1}I} \end{cases}$$

5.2.2 Terminaison de la complétion normalisée

La preuve de la terminaison se fait de la façon suivante : supposons que la complétion ne termine pas, et appliquons les théorèmes précédents (on vérifie aisément que les règles Orientation et Déduction préservent G_F). D'après le théorème 5.6, si la complétion ne termine pas, il y a un symbole AC $+$ tel qu'il y ait une infinité de règles dont le symbole de tête du membre gauche est $+$. Alors nous avons une suite infinie de règles $l_1 = l_{1,1} + \cdots + l_{1,k_1} \to r_1, l_2 = l_{2,1} + \cdots + l_{2,k_2} \to r_2, \ldots$ dans R_∞.

Comme $l_{i,j}$ est un sous-terme strict d'un membre gauche d'une règle de R_∞, il est en forme normale pour R_∞/S, par conséquent chaque $l_{i,j}$ est dans l'ensemble fini $G_F(R_\infty)$. Alors, on applique le corollaire 2.11 sur les multi-ensembles $M_1 = \{l_{1,1}, \ldots, l_{1,k_1}\}, M_2 = \{l_{2,1}, \ldots, l_{2,k_2}\}, \ldots$ et on obtient i et j tels que $M_i \subseteq M_j$ (modulo AC), donc l_i est un sous-terme modulo AC de l_j, donc $l_j \to r_j$ est simplifiable, contradiction. □

La terminaison de la complétion dans tous les cas permet de conclure par le théorème suivant :

Théorème 5.7 *Toute théorie close modulo une théorie S qui est réunion disjointe de théories parmi AC1, ACI, AC1I, AC0 et ACN possède un système de réécriture S-normalisée canonique, et en particulier son problème du mot est décidable.*

5.3 Complétion close modulo la théorie des groupes abéliens

Comme ordre total, nous utilisons celui défini au chapitre 2 c'est-à-dire l'ordre *rpo* avec une précédence totale telle que $\mathcal{F} > - > + > 0$, avec statut multi-ensemble pour $+$ et statut lexicographique pour tous les autres symboles.

	modulo AC	modulo AC1	modulo AG
Temps de calcul	9' 53"	8'18"	3"46
nombre d'appels à l'unification AC	170	145	12
nombre de paires critiques	883	840	6
nombre d'appels au filtrage	143800	137736	904

Figure 5.4 : Résultats pratiques : un groupe finiment engendré

5.3.1 Ensembles normalisant modulo les groupes abéliens dans le cas clos

Comme les équations sont closes le couple normalisant se simplifie en :

$$\begin{cases} \Theta_{AG}(u,v) = \emptyset \\ \Psi_{AG}(u,v) = \{nt \to w, -t \to (n-1)t + (-w) \text{ si } n \geq 2\} \text{ où } (n,t,w) = sym(u,v) \end{cases}$$

On retombe bien sûr exactement sur les règles de symétrisation de Le Chenadec [62], mais ici c'est plus général car valable également en présence de symboles non constants. La terminaison avait été démontré par Le Chenadec dans le cas des constantes, et ici notre preuve dans le cas général est analogue, où l'ensemble des constantes est ici remplacé par l'ensemble G_F dont on a déjà montré la finitude dans le cas général.

5.3.2 Terminaison de la complétion AG-normalisée close

La preuve de terminaison est semblable à celle pour AC1 : si la complétion diverge, il y a une infinité de règles de la forme $l_1 + \cdots + l_k \to r$ où l_i est soit de la forme l'_i, soit de la forme $-l'_i$ pour un l'_i dont le symbole de tête n'est pas dans $F = \{+, -, 0\}$ et on conclut de la même façon en raisonnant sur l'ensemble fini $\{s, -s \mid s \in G_F(R_\infty)\}$.

On obtient donc le théorème suivant, qui généralise au cas où il y a des symboles libres non constants le résultat de décidabilité déjà connu du problème du mot dans les groupes abéliens.

Théorème 5.8 *Toute théorie équationnelle AG-close possède un système de réécriture AG-normalisée convergent, et en particulier son problème du mot est décidable.*

Exemple. La figure 5.5 donne le script de la complétion d'un groupe abélien finiment présenté par trois générateurs a, b et c et les relations $-3a + 2b + 3c = 0$ et $2a + 2b - 2c = 0$. Le système canonique obtenu est constitué des règles des groupes abéliens et des trois règles closes $a \to 4b + c$, $10b \to 0$ et $-b \to 9b$, montrant en particulier que le groupe est isomorphe à $\mathbb{Z}/_{10\mathbb{Z}} \times \mathbb{Z}$.

La figure 5.4 donne les résultats obtenus pour ce même groupe modulo AC, AC1 et AG. On constate sur cet exemple le gain en efficacité spectaculaire lorsqu'on utilise la complétion modulo les groupes abéliens. Cela est bien entendu dû au couple normalisant optimisé, et le gain en efficacité est équivalent au gain en efficacité obtenu par Le Chenadec.

Source	Script de la complétion

`operators`
`  + : AC`
`  0 : constant`
`  - : unary`
`  x,y : variable`
`  a,b,c : constant`
 	
`theory  AG(+,0,-)`
 	
`axioms`
`  -(a+a+a)+b+b+c+c+c = 0;`
`  a+a+b+b+-(c+c) = 0;`
 	
`order`
`  rpo c>b>a>->0>+ ;`
`  + mul`
 	
`end`
	`Orientation de l'equation a+a+b+b+-(c)+-(c) = 0`
	`en`
	`{c+c -> b+b+a+a,`
	`-(c) -> -(b+b+a+a)+c} (2 regles)`
	`Avec les equations critiques :`
	`{}`
	`Orientation de l'equation b+b+b+b+c+-(a) = 0`
	`en`
	`{c -> -(b)+-(b)+-(b)+-(b)+a} (1 regles)`
	`Avec les equations critiques :`
	`{}`
	`Orientation de l'equation -(a)+b+b+b+b =`
	`              -(b)+-(b)+-(b)+-(b)+-(b)+-(b)+-(a)`
	`en`
	`{b+b+b+b+b+b+b+b+b+b -> 0,`
	`-(b) -> -(0)+b+b+b+b+b+b+b+b+b} (2 regles)`
	`Avec les equations critiques :`
	`{}`
	
	`Resultat :`
	`{-(b) -> b+b+b+b+b+b+b+b+b,`
	`b+b+b+b+b+b+b+b+b+b -> 0,`
	`c -> b+b+b+b+b+b+a} (3 regles)`
	
	`Nombre d'appels au filtrage AC     : 904`
	`Nombre d'appels reussis            : 62 (6%)`
	`Nombre d'appels a l'unification AC : 12`
	`Nombre de solutions trouvees       : 6 (0.50 en moyenne)`
	`Temps utilisateur : 3.460 s`
	`Temps systeme : 0.200 s`

Figure 5.5 : Un groupe abélien finiment engendré modulo $AG(+,0,-)$

5.4 Digression : les bases standard d'idéaux de polynômes

Soit A un anneau et $X_1,\ldots,X_k$ une famille d'indéterminées. Pour définir la notion de base standard, il nous faut d'abord supposer donné un ordre sur les « produits » $X_1^{\alpha_1}\cdots X_k^{\alpha_k}$ vérifiant certaines propriétés.

Définition 5.9 *Un ordre $\geq$ sur les produits $\{X_1^{\alpha_1},\ldots,X_k^{\alpha_k} \mid (\alpha_1,\ldots,\alpha_k)\in\mathbb{N}^k\}$ est dit* admissible *s'il est total et si :*

- *pour tous $\alpha_1,\ldots,\alpha_k$, $X_1^{\alpha_1}\cdots X_k^{\alpha_k}\geq 1$;*
- *pour tous $\gamma_1,\ldots,\gamma_k$, si $X_1^{\alpha_1}\cdots X_k^{\alpha_k} \geq X_1^{\beta_1}\cdots X_k^{\beta_k}$ alors $X_1^{\alpha_1+\gamma_1}\cdots X_k^{\alpha_k+\gamma_k} \geq X_1^{\beta_1+\gamma_1}\cdots X_k^{\beta_k+\gamma_k}$.*

Remarquons que ces propriétés impliquent que l'ordre est noethérien. Le monôme principal *d'un polynôme P de $A[X_1,\ldots,X_k]$ est alors simplement son monôme maximal pour cet ordre admissible, que l'on note $t(P)$* (tête *de P). On note alors également $q(P)=P-t(P)$* (queue *de P).*

Les ordres sur les polynômes introduits au chapitre 2 sont des ordres admissibles. En fait on peut démontrer que de tels ordres peuvent être construits en à l'aide d'une forme linéaire positive arbitraire, composée avec un ordre lexicographique pour assurer la totalité [37].

Nous supposons dorénavant fixé un ordre admissible $\geq$.

Définition 5.10 *Soit I un idéal de $A[X_1, \ldots, X_k]$. Une famille de polynômes $\mathcal{P} = \{P_1, ..., P_n\}$ est une* base standard *(par rapport à $\geq$) de I si*

1. *$\mathcal{P}$ engendre I, c'est-à-dire*
$$I = \{Q_1 P_1 + \cdots + Q_n P_n \mid Q_1, \ldots, Q_n \in A[X_1, \ldots, X_k]\};$$
2. *la famille des monômes principaux $\{t(P_1), \ldots, t(P_n)\}$ engendre l'idéal engendré par les monômes principaux des polynômes de I.*

La construction d'une base standard d'un idéal permet de décider de l'appartenance d'un polynôme donné à cet idéal, et en particulier cela donne des renseignements sur les zéros de cet idéal (si l'idéal est engendré par des polynômes $P_1, \ldots, P_n$, cela donne les solutions du système d'équations $P_1 = 0, \ldots, P_k = 0$).

On peut remarquer que dans le cas d'une seule indéterminée sur un corps, on a facilement une base standard d'un idéal engendré par des polynômes $P_1, \ldots, P_n$, c'est la famille à un élément constituée du *pgcd* de $P_1, \ldots, P_n$, que l'on peut calculer par l'algorithme d'Euclide.

5.4.1 Bases standard sur les corps

On suppose ici que A est un corps, et on utilise le fait que le coefficient dominant de tout polynôme peut être normalisé à 1.

Définition 5.11 *Un polynôme normalisé P engendre une relation $\underset{P}{\longrightarrow}$ définie par $Q \underset{P}{\longrightarrow} R$ si $Q = t(P)Q' + Q''$ et $R = -q(P)Q' + Q''$. Si P est un polynôme quelconque, la relation $\underset{P}{\longrightarrow}$ est la relation engendrée par son normalisé.*

Exemple. Si $P = X^2Y + X + 3$, alors
$$\begin{array}{rcl} 5X^4Y^3 & \underset{P}{\longrightarrow} & 5X^2Y^2(-X-3) \\ & = & -5X^3Y^2 - 15X^2Y^2 \\ & \overset{+}{\underset{P}{\longrightarrow}} & 30XY - 5X + 30 \end{array}$$

On constate sur cet exemple que la relation $\underset{P}{\longrightarrow}$ s'apparente beaucoup avec de la réécriture normalisée, puisqu'avant d'essayer d'appliquer une nouvelle réduction, on distribue pour normaliser le polynôme à réduire.

Définition 5.12 *La paire critique entre deux polynômes dont les têtes sont $X_1^{\alpha_1} \cdots X_k^{\alpha_k}$ et $X_1^{\beta_1} \cdots X_k^{\beta_k}$ est formée par les deux réductions possibles à partir de $X_1^{\max(\alpha_1,\beta_1)} \cdots X_k^{\max(\alpha_k,\beta_k)}$. L'algorithme de Buchberger est alors celui donné dans la figure 5.6.*

Proposition 5.13 (Buchberger 1965) *L'algorithme de Buchberger termine toujours et produit comme résultat une base standard de l'idéal engendré par les polynômes donnés.*

Donnée : un ensemble fini de polynômes F

$T_0 := F$; $G_0 := \emptyset$; $i := 0$; $m := 0$;

boucle

{ orientations et normalisations }
tant que $T_i \neq \emptyset$ **faire**
choisir un polynôme p dans T_i;
$(t, q) :=$ tête et queue de p normalisées par G_i;
si t est nulle
alors $T_{i+1} := T_i - \{p\}$; $G_{i+1} := G_i$; $i := i + 1$;
sinon $K :=$ ensemble des étiquettes des polynômes de G_i simplifiables par (t, q);
$T_{i+1} := (T_i - \{p\}) \cup \{p_k \mid k \in K\}$;
$m := m + 1$;
$G_{i+1} := \{j : (t_j, q'_j) \mid j \notin K\} \cup \{m : (t, q)\}$
avec $q'_j :=$ normalisé de q_j par $G_i \cup \{m : (t, q)\}$ et $m : (t, q)$ non marqué;
$i := i + 1$;
fin tant que;

{ calcul des paires critiques }
si tous les polynômes de G_i sont marqués
alors renvoyer G_i
sinon choisir un polynôme non marqué d'étiquette k dans G_i;
$T_{i+1} :=$ ensembles des paires critiques entre le polynôme numéro k et les polynômes marqués de G_i d'étiquettes inférieures à k;
$G_{i+1} := G_i$ où le polynôme d'étiquette k est maintenant marqué;
$i := i + 1$;

fin boucle;

Figure 5.6 : Algorithmes de Buchberger et de Kandri-Rody et Kapur

5.4.2 Bases standard sur des anneaux qui ne sont pas des corps

Kandri-Rody et Kapur ont généralisé la technique de Buchberger à l'anneau $\mathbb{Z}$ et sur les anneaux euclidiens en général. On suppose donc que A est euclidien, c'est-à-dire intègre ($ab = 0$ implique $a = 0$ ou $b = 0$) et muni d'une division euclidienne, définie par une fonction $\delta : A - \{0\} \mapsto \mathbb{N}$ telle que pour tous $a, b \in A$, $b \neq 0$, il existe $q, r \in A$ tels que $a = bq + r$ et $r = 0$ ou $\delta(r) < \delta(b)$.

Exemple. Sur $\mathbb{Z}$, on peut prendre $\delta(x) = |x|$. Notons qu'il n'y a pas unicité du quotient et du reste, par exemple pour diviser -1 par 3 : $-1 = 3 \times (-1) + 2$ et $-1 = 3 \times 0 + (-1)$.

Afin de définir de façon unique la réduction engendrée par un polynôme, il nous faut supposer fixée une fonction de division euclidienne div donnant un quotient et un reste.

Définition 5.14 *Un polynôme P dont la tête est $aX_1^{\alpha_1}\cdots X_k^{\alpha_k}$ et la queue est P' engendre une relation $\underset{P}{\longrightarrow}$ définie par $Q \underset{P}{\longrightarrow} R$ si*

- $Q = bX_1^{\beta_1}\cdots X_k^{\beta_k} + Q''$;
- *pour tout i, $\beta_i \geq \alpha_i$;*
- $(q,r) = \mathrm{div}(a,b)$ *avec* $q \neq 0$;
- $R = rX_1^{\beta_1}\cdots X_k^{\beta_k} - qP'X_1^{\beta_1-\alpha_1}\cdots X_k^{\beta_k-\alpha_k} + Q''$.

Exemple. Si $P = 2X^2Y + X + 3$, alors

$$\begin{array}{rcl} 5X^4Y^3 & \underset{P}{\longrightarrow} & 2\times(2X^2Y^2(-X-3)) + X^4Y^3 \\ & = & X^4y^3 - 2X^3Y^2 - 6X^2Y^2 \\ & \overset{+}{\underset{P}{\longrightarrow}} & X^4Y^3 + X^2Y + 6XY + 9 \end{array}$$

On constate encore sur cet exemple que la relation $\underset{P}{\longrightarrow}$ s'apparente avec de la réécriture normalisée.

Définition 5.15 *La paire critique entre deux polynômes dont les têtes sont $aX_1^{\alpha_1}\cdots X_k^{\alpha_k}$ et $bX_1^{\beta_1}\cdots X_k^{\beta_k}$ est formée par les deux réductions possibles à partir de*

$$cX_1^{\max(\alpha_1,\beta_1)}\cdots X_k^{\max(\alpha_k,\beta_k)}$$

où $c = \mathrm{ppcm}(a,b)$. L'algorithme de Kandri-Rody et Kapur est alors celui donné dans la figure 5.6 (en fait le même que pour Buchberger, seuls la notion de réduction et de paire critique change).

Proposition 5.16 (Kandri-Rody Kapur 1984) *Si A est un anneau euclidien, tout idéal possède une base standard, calculée par l'algorithme de Kandri-Rody et Kapur.*

En fait, on peut généraliser le cas des anneaux euclidiens au cas des anneaux qui possèdent déjà un algorithme de calcul de bases standard, puisque comme on l'a déjà remarqué les algorithmes de calcul de bases standard sont des généralisations de l'algorithme d'Euclide de calcul du *pgcd* [37].

5.4.3 Une application à un problème du mot

Une application qui nous concerne plus particulièrement est la décidabilité du problème du mot dans les théories CR-closes lorsque la signature ne contient que des constantes comme symboles (autres que ceux de CR). En effet, un ensemble d'équations closes définissant une telle théorie peut être considéré comme une ensemble d'équations entre polynômes à coefficients dans $\mathbb{Z}$, dont les indéterminées sont les constantes libres de la signature. $\mathbb{Z}$ étant euclidien, l'algorithme de Kandri-Rody et Kapur permet de calculer une base standard et donc donne un algorithme de décision du problème du mot.

Nous allons voir dans la section suivante que la complétion normalisée close modulo CR généralise cette technique de résolution du problème du mot au cas où il y a des symboles libres non constants dans la signature. Le fait intéressant est que nous pourrons alors aller en sens inverse et montrer que la complétion normalisée modulo CR permet de calculer des bases standard.

5.5 Complétion close modulo les anneaux commutatifs

Dans cette section, nous montrons la terminaison de la complétion normalisée modulo CR lorsque les équations données sont closes, et nous obtenons en conséquence un algorithme de calcul de base standard sur des polynômes à coefficients dans $\mathbb{Z}$, puis ensuite sur les corps finis $\mathbb{F}_q$.

5.5.1 Couple normalisant pour les anneaux commutatifs dans le cas clos

Comme ordre AC total, nous utilisons l'ordre total permettant d'orienter dans le bon sens les règles précédentes (en particulier la distributivité) décrit en 2.5.4.

Nous utilisons les ensembles critiques suivants:

$$\begin{cases} \Theta_{CR}(u,v) & = & \emptyset \\ \Psi_{CR}(u,v) & = & \{nt \to w\} \cup \\ & & \{n(t*x) \to w*x, -t \to (n-1)t + (-w), n(t*x) \to w*x \text{ si } n \geq 2\} \\ & \text{où } (n,t,w) = sym(u,v) & \end{cases}$$

Exemple. Si $a > b$, on a par exemple $\Psi_{CR}(2(a*a) - 3(b*b*b), -a+4b)$ qui contient les quatre équations

$$\begin{cases} 2(a*a) \to -a + 3(b*b*b) + 4b \\ 2(a*a*x) \to -a*x + 3(b*b*b*x) + 4(b*x) \\ -(a*a) \to a*a + a - 3(b*b*b) - 4b \\ -(a*a*x) \to a*a*x + a*x - 3(b*b*b*x) - 4(b*x) \end{cases}$$

5.5.2 Terminaison

Les algorithmes de Buchberger et de Kandri-Rody et Kapur terminent quelles que soient les données. Nous montrons maintenant que c'est aussi le cas pour notre algorithme de complétion : la démonstration est encore similaire aux cas précédents des groupes et de AC1I. Il y a cependant un point important dans le cas des anneaux, c'est qu'il y a introduction de variables, donc l'ensemble des règles ne reste pas clos. Néanmoins on peut voir que les équations déduites de ces règles non closes sont, elles, closes. On est sûr alors que les règles non-closes sont uniquement celles de la forme donnée dans Ψ_{CR}, et donc nous ne pouvons pas avoir d'équation inorientable. Pour finir la démonstration de la terminaison, on sait par le théorème 5.6 que si la complétion ne termine pas, il y a une infinité de règles dont le symbole de tête du membre gauche est $+$, $-$ ou $*$. Il nous faut maintenant regarder plus précisément la forme des règles produites par la complétion : on voit alors aisément qu'il y a une infinité de règles de la forme $n(u*\cdots*u) \to \ldots$ où u est dans $G_{+,-,*,0,1}(R_\infty)$ et on conclut par le corollaire 2.11 comme pour les groupes. On a donc le théorème suivant, généralisant le résultat de Kandri-Rody et Kapur si l'on est en présence de symboles non constants.

Théorème 5.17 *Toute théorie équationnelle CR-close possède un système de réécriture CR-normalisée convergent, et en particulier son problème du mot est décidable.*

Exemple. Pour calculer une base standard de l'idéal $(2X^2Y - Y, 3XY^2 - X)$ on donne comme donnée à la complétion l'ensemble d'équations $\{2(X*X*Y) - Y = 0, 3(X*Y*Y) - X = 0\}$

Source	Script de la complétion

`operators`
`  +,* : AC`
`  0,1 : constant`
`  -   : unary`
`  X,Y,Z : constant`
`  x,y,z : variable`
 	
`theory  CR(+,0,-,*,1)`
 	
`axioms`
`  (X*X*Y) + (X*X*Y) +`
`    -(Y) = 0;`
`  (X*Y*Y) + (X*Y*Y) +`
`    (X*Y*Y) + -(X) = 0;`
 	
`order`
`  rpo Y>X>1>*>->0>+ ;`
`      + mul, * mul`
 	
`end`
	`Resultat :`
	`{(x*Y*Y)+(x*Y*Y)+(x*Y*Y) -> (x*X*X)+(x*X*X),`
	`(Y*Y)+(Y*Y)+(Y*Y) -> (X*X)+(X*X),`
	`(X*X*Y)+(X*X*Y) -> Y,`
	`-(X*X*X) -> -(X)+(X*X*X),`
	`-(X*X*Y) -> -(Y)+(X*X*Y),`
	`(x*X*X*Y)+(x*X*X*Y) -> x*Y,`
	`-(x*X*X*X) -> -(x*X)+(x*X*X*X),`
	`X*X*Y*Y -> (Y*Y)+(Y*Y)+-(X*X),`
	`-(x*X*X*Y) -> -(x*Y)+(x*X*X*Y),`
	`(x*X*X*X)+(x*X*X*X) -> x*X,`
	`(X*X*X)+(X*X*X) -> X,`
	`-(Y*Y) -> -(X*X)+-(X*X)+(Y*Y)+(Y*Y),`
	`-(x*Y*Y) -> -(x*X*X)+-(x*X*X)+(x*Y*Y)+(x*Y*Y)} (13 regles)`
	
	`Nombre d'appels au filtrage AC     : 19422`
	`Nombre d'appels reussis            : 585 (3%)`
	`Nombre d'appels a l'unification AC : 180`
	`Nombre de solutions trouvees       : 117 (0.65 en moyenne)`
	`Temps utilisateur : 59.310 s`
	`Temps systeme : 0.730 s`

Figure 5.7 : Calcul d'une base standard

où X, Y sont deux constantes, $Y > X$ dans la précédence. Le script de la complétion est donné figure 5.7.

Le résultat correspond à la base standard $\{2X^2Y - Y, X^2Y^2 - X^2 + Y^2, 3Y^2 - 2X^2, 2X^3 - X\}$.

Il faut quand même remarquer que du point de vue de l'efficacité, on est loin de rivaliser avec les programmes dédiés au calcul de bases standard, mais c'est tout à fait normal puisque ces derniers n'utilisent pas l'unification AC.

Il serait intéressant de voir l'amélioration que pourrait donner l'utilisation de l'unification AC1 plutôt que AC, mais malheureusement cela n'est pas encore implanté.

On peut remarquer également que l'on peut calculer des bases standard non seulement sur $\mathbb{Z}$ mais aussi sur n'importe quel extension algébrique puisqu'un tel quotient est précisément le quotient d'un anneau de polynômes par un idéal, par exemple l'anneau des entiers de Gauss : $\mathbb{Z}[i] = \mathbb{Z}[X]/_{(X^2+1)}$.

5.5.3 Cas des corps premiers $\mathbb{Z}_p$

On ne peut pas directement plonger l'algorithme de calcul de bases standard sur les corps dans la complétion normalisée car la théorie des corps ne possède de présentation équationnelle. On peut contourner le problème pour le cas des corps premiers de caractéristique non nulle p, comme cela a été remarqué par Bündgen [21]. Le problème du calcul de base standard sur $\mathbb{Q}$ par la complétion reste donc ouvert : existe-t-il un système convergent S pour lequel la complétion S-normalisée permet de calculer des bases standard de polynômes à coefficient dans $\mathbb{Q}$?

En effet, si on ajoute l'équation définissant la caractéristique, qui est $px = 0$, à la présentation des anneaux commutatifs, on obtient alors un système convergent qui est le suivant :

$$FP(p) \left\{ \begin{array}{l} x + 0 \to x \\ -x \to \underbrace{x + \cdots + x}_{p-1} \\ \underbrace{x + \cdots + x}_{p} \to 0 \\ x * 1 \to x \\ x * (y + z) \to (x * y) + (x * z) \\ x * 0 \to 0 \end{array} \right.$$

On remarque que l'ensemble des formes normales closes pour ce système est exactement $\{0, 1, \ldots, \underbrace{1 + \cdots + 1}_{p-1}\}$ qu'on peut mettre naturellement en bijection avec $\mathbb{Z}_p$, et bien sûr cette bijection est alors un morphisme d'anneaux. Comme d'autre part on a une opération d'inverse sur $\mathbb{Z}_p$, on a une opération d'inverse analogue sur l'ensemble des formes normales et cette opération d'inverse peut être codée par $x^{-1} = x^{p-2}$. Cette remarque conduit à la définition suivante de couple normalisant :

$$\left\{ \begin{array}{l} \Psi_{FP(p)}(u, v) = \{s \to (\alpha^{p-2} \bmod p)t\} \text{ où } (\alpha s, t) = sym(u, v) \\ \Theta_{FP(p)}(u, v) = \Theta_{CR} \cup \Theta_{AC1N(+,0)} \end{array} \right.$$

La terminaison est dans ce cas plutôt plus simple à montrer que dans le cas des anneaux puisqu'ici il n'y a pas d'introduction de variables.

Théorème 5.18 *Toute théorie équationnelle FP-close possède un système de réécriture FP-normalisée convergent, et en particulier son problème du mot est décidable.*

5.5.4 Extension à tous les corps finis

Si F est un corps fini arbitraire de cardinal q, si p est sa caractéristique alors $q = p^r$ pour un certain entier r. De plus, F est alors isomorphe à $\mathbb{Z}_p[X]/(\phi(X))$ où ϕ est un certain polynôme (un polynôme minimal d'une racine $q^{\text{ème}}$de l'unité). On a alors deux solutions pour calculer des bases standard pour des polynômes à coefficients dans F, soit on calcule une base standard sur $\mathbb{Z}_p$ en y adjoignant ϕ, soit on complète par la complétion normalisée modulo le nouveau système (a est une nouvelle constante correspondant à l'indéterminée X) :

$$FP(q) \left\{ \begin{array}{l} x + 0 \to x \\ -x \to \underbrace{x + \cdots + x}_{p-1} \\ \underbrace{x + \cdots + x}_{p} \to 0 \\ x * 1 \to x \\ x * (y + z) \to (x * y) + (x * z) \\ x * 0 \to 0 \\ a^r \to -q(\phi(a)) \end{array} \right.$$

Le couple normalisant est le même qu'avant, avec α^{q-2} dans la définition de Ψ.

<table>
<tr><th>Source</th><th>Script de la complétion</th></tr>
<tr><td><pre>operators
  +,* : AC
  0,1 : constant
  -   : unary
  X,Y,Z : constant
  x,y,z : variable

theory FP(5,+,0,-,*,1)

axioms
  (X*X*Y) + (X*X*Y) +
    -(Y) = 0;
  (X*Y*Y) + (X*Y*Y) +
    (X*Y*Y) + -(X) = 0;

order
  rpo Y>X>1>*>->0>+ ;
      + mul, * mul

end</pre></td><td><pre>Resultat :
{X*X*X -> X+X+X,
Y*Y -> (X*X)+(X*X)+(X*X)+(X*X),
X*X*Y -> Y+Y+Y} (3 regles)

Nombre d'appels au filtrage AC   : 4828
Nombre d'appels reussis   : 313 (6%)
Nombre d'appels a l'unification AC : 28
Nombre de solutions trouvees   : 16 (0.57 en moyenne)
Temps utilisateur : 27.820 s
Temps systeme : 0.510 s</pre></td></tr>
</table>

Figure 5.8 : Calcul d'une base standard sur le corps premier $\mathbb{F}_5$

Exemple. L'exemple le plus simple de corps fini non premier est le corps à quatre éléments $\mathbb{F}_4$, qui est isomorphe à $\mathbb{Z}_2[X]/(X^2 + X + 1)$. Ainsi, pour calculer une base standard à partir d'une famille $\mathcal{P}$ de polynômes à coefficients dans $\mathbb{F}_4$ (donnés en fonction de 1 et a), on peut soit calculer une base standard de $\mathcal{P} \cup \{a^2 + a + 1\}$ sur $\mathbb{Z}_2$ où a est considéré comme une nouvelle indéterminée, plus petite que les autres dans l'ordre; soit compléter modulo le système convergent suivant :

$$FP(4) \quad \begin{cases} x + 0 \to x \\ -x \to x \\ x + x \to 0 \\ x * 1 \to x \\ x * (y + z) \to (x * y) + (x * z) \\ x * 0 \to 0 \\ a^2 \to a + 1 \end{cases}$$

Exemple. Un calcul de base standard sur $\mathbb{F}_p$. On peut par exemple essayer de calculer la base standard sur $\mathbb{F}_5$ de l'idéal engendré par les mêmes polynômes que notre premier exemple sur $\mathbb{Z}$. Le résultat est donné figure 5.8. Évidemment le résultat est très différent.

5.6 Anneaux booléens

Pour terminer ce chapitre, nous nous intéressons au cas de la théorie des anneaux booléens. Nous allons voir en fait que les choses se passent plus simplement que dans le cas des anneaux commutatifs. En fait, cela correspond à calculer des bases standard par l'algorithme pour les anneaux commutatifs en ajoutant à l'ensemble de polynômes de départ $X_i^2 \to X_i$ pour toutes les indéterminées X_i.

Le système convergent que l'on considère est

$$BR \left\{ \begin{array}{l} x + 0 \to x \\ -x \to x \\ x + x \to 0 \\ x * 1 \to x \\ x * (y + z) \to (x * y) + (x * z) \\ x * 0 \to 0 \\ x * x \to x \end{array} \right.$$

Comme ordre AC total, nous utilisons le même ordre que pour les anneaux commutatifs.

5.6.1 Couple normalisant pour les anneaux booléens

Nous utilisons les ensembles critiques suivants:

$$\left\{ \begin{array}{lcl} \Theta_{BR}(u, v) & = & \emptyset \\ \Psi_{BR}(u, v) & = & \{t \to w\} \\ & & \text{où } (1, t, w) = sym(u, v) \end{array} \right.$$

On constate ici que c'est beaucoup plus simple que pour les anneaux commutatifs arbitraires puisque le coefficient dominant est toujours 1, du fait de la présence de la règle $x + x \to 0$.

Une remarque intéressante ici est que le calcul peut se faire avec l'unification booléenne, car elle est connue comme ayant de bonnes propriétés [18].

5.6.2 Terminaison

Pour le cas des anneaux booléens, la preuve de terminaison n'est qu'un cas particulier très simple du cas des anneaux commutatifs.

Théorème 5.19 *Toute théorie équationnelle BR-close possède un système de réécriture BR-normalisée convergent, et en particulier son problème du mot est décidable.*

Chapitre 6

Résultats d'indécidabilité

Dans ce chapitre, nous nous intéressons à des résultats d'indécidabilité concernant le problème du mot dans diverses classes de théories équationnelles E-closes. Nous rappelons d'abord des résultats connus maintenant depuis quelques dizaines d'années, et nous montrons également un résultat nouveau, l'indécidabilité du problème du mot dans les théories ACD-closes. L'outil de base pour montrer ces résultats d'indécidabilité consiste à réduire le problème de l'arrêt d'une machine de Turing arbitraire, dont l'indécidabilité est connue depuis 1936 [87, 88], en un problème du mot pour une théorie E-close.

6.1 Machines de Turing

Nous prenons ici une définition simple des machines de Turing, celle donnée dans le *Handbook of Mathematical Logic* [12] : une machine de Turing est déterministe, a un ruban infini dans les deux sens, et le vocabulaire est fixé à $\{0,1\}$. Nous n'aurons pas besoin des variantes non déterministes, ni celles avec plusieurs rubans, ni d'utiliser un vocabulaire de bande plus général que $\{0,1\}$.

Définition 6.1 *Une* machine de Turing *est un triplet $\langle Q, q_0, \delta\rangle$ où Q est un ensemble fini (l'ensemble des états), $q_0 \in Q$ (l'état initial), et δ (la fonction de transition) est une fonction partielle de $Q \times \{0,1\}$ dans $Q \times \{0,1\} \times \{L, R\}$.*

Définition 6.2 *Une* configuration *de la machine de Turing $\mathcal{M} = \langle Q, q_0, \delta\rangle$ est un triplet $\langle T, p, q\rangle$ où T (le ruban) est une application de $\mathbb{Z}$ dans $\{0,1\}$, $p \in \mathbb{Z}$ (la position courante) et $q \in Q$ (l'état courant).*

Dans la suite, nous utilisons les lettres grecques α et β pour noter des chiffres binaires et nous notons le ruban T comme un mot infini $\ldots\alpha_{-1}\alpha_0\alpha_1\alpha_2\ldots$ si $T(p) = \alpha_p$.

Définition 6.3 $\mathcal{M}$ transforme *la configuration $\langle T, p, q\rangle$ en la configuration $\langle T', p', q'\rangle$ si et seulement si $\delta(q, T(p)) = (q', \alpha, d)$, $T'(p) = \alpha$, $\forall x \neq p\ T'(x) = T(x)$, et $p' = p - 1$ si $d = L$, $p' = p + 1$ si $d = R$.*

Nous notons alors $\langle T, p, q\rangle \underset{\mathcal{M}}{\Rightarrow} \langle T', p', q'\rangle$.

Intuitivement, $\delta(q, \alpha) = (q', \beta, d)$ signifie que si $\mathcal{M}$ est dans l'état q et α est le symbole à la position courante sur le ruban, alors β est écrit sur le ruban, l'état courant devient q', et la position courante est décalée dans la direction indiquée par d, L pour gauche et R pour droite. Si $\delta(q,\alpha)$ n'est pas défini, alors la machine s'arrête.

Définition 6.4 *Le* calcul *de $\mathcal{M}$ sur la donnée initiale $\alpha_1 \dots \alpha_n$ est la suite finie ou infinie de configurations obtenues à partir de la configuration initiale $\langle T_0, 0, q_0\rangle$ où $T_0 = \dots 00\alpha_1 \dots \alpha_n 00 \dots$ ($T_0(0) = \alpha_1$). $\mathcal{M}$* s'arrête *si la suite est finie.*

Définition 6.5 *Le* problème de l'arrêt *d'une machine de Turing $\mathcal{M}$ est :*

Donnée	:	*un mot $\alpha_1 \dots \alpha_n$.*
Question	:	*est-ce que le calcul de $\mathcal{M}$ sur $\alpha_1 \dots \alpha_n$ s'arrête ?*

Pour un entier naturel $n \in \mathbb{N}$, nous notons $\overline{n}$ le mot $\underbrace{1\dots1}_{n+1}$.

Définition 6.6 *Une fonction partielle k-aire $\mathbb{N}^k \mapsto \mathbb{N}$ est dite* récursive *s'il existe une machine de Turing $\mathcal{M}$ telle que si on fait calculer $\mathcal{M}$ sur la donnée initiale $\overline{x_1}0\overline{x_2}0\dots0\overline{x_k}$ alors*

- *si $f(x_1,\dots,x_k)$ est défini alors $\mathcal{M}$ s'arrête avec comme position courante le 1 le plus à gauche de $0\overline{f(x_1,\dots,x_k)}0$.*
- *si $f(x_1,\dots,x_k)$ n'est pas défini alors $\mathcal{M}$ ne s'arrête pas.*

Définition 6.7 *Un ensemble de k-uplets $E \subset \mathbb{N}^k$ est dit* récursif *si sa fonction caractéristique $\chi_E : \mathbb{N}^k \mapsto \{0,1\}$ est récursive.*

Le théorème suivant, dû à Turing en 1936, montre qu'il existe une machine de Turing dont le problème de l'arrêt est indécidable.

Théorème 6.8 *Il existe une fonction récursive f telle que le domaine de définition de f n'est pas récursif, autrement dit le problème de l'arrêt de la machine calculant f est indécidable.*

Ce théorème signifie encore qu'il n'y a pas d'algorithme pour décider si la machine de Turing de f s'arrête pour une entrée donnée.

6.2 Indécidabilité du problème du mot A-clos

Nous donnons dans cette section la démonstration classique d'indécidabilité du problème du mot pour une théorie close modulo associativité [12]. La démonstration de l'indécidabilité du problème du mot ACD-clos, qui sera donnée dans la section suivante, sera largement inspirée de celle-ci.

6.2.1 La preuve d'indécidabilité

Nous donnons celle du *Handbook of mathematical logic* [23] : l'idée est relativement simple, on associe à une machine de Turing arbitraire $\mathcal{M}$ un ensemble d'équations closes $E_{\mathcal{M}}$ telles que le problème de l'arrêt de $\mathcal{M}$ soit équivalent à un problème du mot pour $E_{\mathcal{M}}$. Le ruban de la machine est codé par un mot formé de 0 et de 1, plus un symbole spécial h aux extrémités pour représenter une infinité de 0 de chaque côté, et enfin un symbole donnant l'état courant de la machine est placé à gauche du chiffre se trouvant à la position de la tête de lecture.

Définition 6.9 *Soit $\mathcal{M} = \langle Q, q_0, \delta \rangle$ un machine de Turing arbitraire. On associe à cette machine un ensemble de règles $R_{\mathcal{M}}$ sur la signature $Q \cup \{0, 1, q_f, h\}$ donné par :*

- *pour chaque transition $\delta(q, \alpha) = (q', \beta, R)$*

$$\begin{cases} q\alpha 0 \to \beta q' 0 \\ q\alpha 1 \to \beta q' 1 \\ q\alpha h \to \beta q' 0h \end{cases}$$

- *pour chaque transition $\delta(q, \alpha) = (q', \beta, L)$*

$$\begin{cases} 0q\alpha \to q' 0\beta \\ 1q\alpha \to q' 1\beta \\ hq\alpha \to hq' 0\beta \end{cases}$$

- *pour chaque transition $\delta(q, \alpha)$ non définie*

$$q\alpha \to q_f$$

- *et enfin des règles spéciales destinées à « envoyer » tout état d'arrêt sur q_f*

$$\begin{cases} q_f 0 \to q_f \\ q_f 1 \to q_f \\ 0q_f h \to q_f h \\ 1q_f h \to q_f h \\ hq_f h \to q_f \end{cases}$$

Le lemme suivant montre l'adéquation du codage.

Lemme 6.10 *La machine $\mathcal{M}$ s'arrête sur la donnée $\alpha_1 \ldots \alpha_n$ si et seulement si*

$$hq_0\alpha_1 \ldots \alpha_n h \xrightarrow[R_{\mathcal{M}}]{*} q_f$$

Définition 6.11 *L'ensemble d'équations $E_{\mathcal{M}}$ est $\{u = v \mid u \to v \in R_{\mathcal{M}}\}$.*

Pour obtenir la proposition suivante, il suffit de remarquer que $R_{\mathcal{M}}$ est sans paire critique sur des mots ne contenant qu'un symbole de Q.

Proposition 6.12 *La machine $\mathcal{M}$ s'arrête sur la donnée $\alpha_1 \ldots \alpha_n$ si et seulement si*

$$hq_0\alpha_1 \ldots \alpha_n h \xleftrightarrow[E_{\mathcal{M}}]{*} q_f$$

On en déduit alors facilement le théorème recherché.

Théorème 6.13 *Il existe un ensemble d'équations closes E tel que le problème du mot de $\mathrm{A} \cup E$ est indécidable.*

6.2.2 Quelques cas pathologiques

S'il est maintenant établi qu'il existe des théories A-closes avec problème du mot indécidable, il reste quand même une question intéressante : existe-il des théories A-closes dont le problème du mot soit décidable mais qui n'admette pas pour autant de système canonique ? Cette question à été résolue tout d'abord par Kapur et Narendran [52] qui ont donné l'exemple suivant à une seule règle $\{aba = bab\}$. La question n'est pas encore complètement résolue car il y a une astuce : il y a un système canonique si l'on a le droit de rajouter un symbole c :

$$\begin{cases} ab \to c \\ ca \to bc \\ bcb \to cc \\ ccb \to acc \end{cases}$$

(on montre la terminaison par composée lexicographique de la longueur, puis d'un *rpo* avec précédence $c > b > a > .$ et . à statut lexicographique gauche-droite, puis on vérifie la confluence locale)

Ensuite, Squier a donné l'exemple suivant, pour une signature à cinq constantes {a,b,c,d,d'} :

$$\begin{cases} ab = a \\ da = ac \\ d'a = ac \end{cases}$$

et a montré que cette théorie n'admet pas de système canonique même si l'on rajoute un nombre fini arbitraire de symboles.

La preuve est intéressante car elle utilise des notions d'algèbre assez subtiles, inspirées de la géométrie algébrique : les groupes d'homologie. Ils sont invariants par isomorphisme, et permettent de lire des propriétés du monoïde considéré.

Il serait hors de propos ici de donner tous les détails et nous seulement un canevas de la démonstration [58]. À un monoïde M on associe une suite de groupes $(H_n(M))_{n\geq 1}$ dits *groupes d'homologies* de M. Si M et M' sont isomorphes, alors pour tout n, $H_n(M)$ et $H_n(M')$ sont isomorphes. Voici quelques exemple de propriétés que l'on peut « lire » sur les groupes d'homologie :

- si M est le monoïde trivial, alors pour tout n, $H_n(M) = 0$;
- si M est finiment engendré, alors $H_1(M)$ est finiment engendré;
- si M est libre, alors pour tout $n \geq 2$, $H_n(M) = 0$;
- si M est engendré par un ensemble fini d'équations, alors $H_2(M)$ est finiment engendré.

Mais la propriété qui nous intéresse est celle énoncée par le théorème suivant.

Théorème 6.14 (Squier 1987) *Si M admet une présentation par un système canonique fini alors $H_3(M)$ est finiment engendré.*

Corollaire 6.15 *Le monoïde M présenté par les équations*

$$\begin{cases} ab = a \\ da = ac \\ d'a = ac \end{cases}$$

n'admet pas de système canonique fini.

Démonstration. On remarque que M admet le système canonique infini

$$\begin{cases} (A_n) & ac^nb \to ac^n \\ (B) & da \to ac \\ (B') & d'a \to ac \end{cases}$$

et $H_3(M)$ est $\mathbb{Z}[(P_n - P'_n)_{n\geq 1}]$, où P_n est la paire critique entre A_n et B, et P'_n celle entre A_n et B'. $H_3(M)$ est donc de dimension infinie. □

6.3 Indécidabilité du problème du mot G-clos

L'indécidabilité du problème du mot G-clos découle de l'indécidabilité du problème du mot dans les groupes finiment engendrés, dont la démonstration est nettement plus complexe que la précédente. La raison vient de la propriété de *régularité* : si une équation $ww_1 = ww_2$ est valide, alors l'équation $w_1 = w_2$ aussi. Si l'on regarde les équations qui codent les machines de Turing dans le cas des semi-groupes, on a par exemple l'équation $q\alpha 0 = \beta q' 0$ qui implique donc, si l'on est dans un groupe, $q\alpha = \beta q'$. Ce phénomène se reproduit quasiment pour toutes les autres équations, et finalement on ne code plus du tout le comportement de la machine !

La première démonstration d'indécidabilité date de 1955 et est due à Novikov [77], et une autre preuve a été obtenue indépendamment par Boone [14]. Ces démonstrations ont été largement améliorées et écourtées depuis, mais il serait hors de propos de donner ici une transcription complète de la preuve. nous conseillons de se reporter à l'article de synthèse de Stillwell [85].

Pour ce qui est de l'existence de cas pathologiques comme dans le cas A-clos, de théorie avec problème du mot décidable mais sans système canonique, Squier a également donné de tels exemples [84]

6.4 Indécidabilité du problème du mot ACD-clos

Étant donnée une machine de Turing, nous allons coder son problème de l'arrêt en un problème du mot pour une certaine théorie ACD-close.

Notre méthode est similaire à la démonstration d'indécidabilité du problème du mot A-clos, mais dans cette démonstration une configuration de la machine était codée par un mot sur un certain alphabet, or ici nous n'avons pas d'opérateur qui soit associatif mais non commutatif, donc nous ne pouvons pas utiliser des mots. En fait, la propriété importante des mots est que ce sont des *séquences ordonnées* de lettres. Ici nous allons utiliser des *polynômes*, où les coefficients sont des lettres, pour coder des séquences ordonnées. Les polynômes se codent de façon naturelle avec deux opérateurs AC avec distributivité.

6.4.1 Représentation d'une configuration par un polynôme

Soit $\mathcal{M} = \langle Q, q_0, \delta\rangle$ une machine de Turing arbitraire, avec $Q = \{q_0, \ldots, q_N\}$. Nous donnons d'abord l'alphabet pour coder $\mathcal{M}$:

Définition 6.16 *Soit $Q' = \{q'_0, \ldots, q'_N\}$ une « copie » de Q. La signature $\mathcal{F}(\mathcal{M})$ (ou $\mathcal{F}$ s'il n'y a pas d'ambiguïté) est*

$$Q \cup Q' \cup \{q, q', q'', h, h', X, I, U, 0, 1\} \cup \{*, +\}$$

où tous les symboles autres que $*$ et $+$ sont des constantes.

Définition 6.17 *Une configuration $\langle T, p, q_j \rangle$ où $T = \ldots 00\alpha_1 \ldots \alpha_n 00 \ldots$ et $T(p) = \alpha_i$ pour un certain i est codé par le terme suivant (un polynôme en X, le symbole $*$ est implicite) :*

$$\overline{\langle T, p, q_j \rangle} = h + X\alpha_1 + \ldots + X^{i-1}\alpha_{i-1} + X^i q_j + X^{i+1}\alpha_i + \ldots + X^{n+1}\alpha_n + X^{n+2}h'$$

Les constantes q et q' vont simuler un état d'arrêt, h et h' sont les bornes du ruban (ils représentent une infinité de 0). I et U vont permettre d'ajouter un 0 à gauche du ruban si nécessaire.

Il faut nous assurer qu'il y a toujours un 0 ou un 1 à lire à droite du caractère q_j, c'est-à-dire que si i est l'exposant du monôme dont le coefficient est q_j, alors il y a un et un seul monôme $X^{i+1}0$ ou $X^{i+1}1$. Cela signifie que si l'on se déplace trop loin à gauche ou à droite, il faut ajouter un 0 au ruban. L'ajout d'un 0 à droite est trivial car seul l'exposant h' doit être changé, mais pour ajouter un 0 à gauche, il faut incrémenter de un tous les exposants. Nous allons décrire la méthode dans la section suivante.

6.4.2 Simulation du comportement de la machine par un ensemble de règles de réécriture

Le comportement de la machine de Turing est simulé par de la réécriture ACD-close (non normalisée) avec un ensemble de règles closes définies de façon unique par les transitions de la machine. Remarquons que la réécriture modulo ACD est décidable car les classes de congruences modulo ACD sont finies, mais de toute façon cet argument n'est pas utile à la preuve d'indécidabilité, puisque nous intéressons en fait à la théorie équationnelle engendrée par ACD et les règles.

Définition 6.18 *$\mathcal{M}$ définit un ensemble de règles closes $R(\mathcal{M})$ de la façon suivante :*

- *pour chaque transition $\delta(q_i, \alpha) = (q_j, \beta, R)$, $R(\mathcal{M})$ contient*

$$q_i + X\alpha + X^2 0 \rightarrow \beta + Xq_j + X^2 0 \quad \text{(a)}$$
$$q_i + X\alpha + X^2 1 \rightarrow \beta + Xq_j + X^2 1 \quad \text{(b)}$$
$$q_i + X\alpha + X^2 h' \rightarrow \beta + Xq_j + X^2 0 + X^3 h' \quad \text{(c)}$$

- *pour chaque transition $\delta(q_i, \alpha) = (q_j, \beta, L)$, $R(\mathcal{M})$ contient*

$$0 + Xq_i + X^2\alpha \rightarrow q_j + X0 + X^2\beta \quad \text{(d)}$$
$$1 + Xq_i + X^2\alpha \rightarrow q_j + X1 + X^2\beta \quad \text{(e)}$$
$$h + Xq_i + X^2\alpha \rightarrow h + Xq'_j + X^2 0 + X^3 I\beta \quad \text{(f)}$$

- *pour chaque $\delta(q_i, \alpha)$ indéfini*

$$q_i + X\alpha \rightarrow q + X\alpha \quad \text{(g)}$$

- *pour tout* $\alpha, \beta \in \{0,1\}$, $R(\mathcal{M})$ *contient*

$$\begin{array}{rcll} I\alpha + \beta & \to & \alpha + XI\beta & \text{(h)} \\ I\alpha + h' & \to & U\alpha + Xh' & \text{(i)} \\ \alpha + UX\beta & \to & U\alpha + X\beta & \text{(j)} \\ q'_j + UX\beta & \to & q_j + X\beta & \text{(k)} \end{array}$$

Ces règles effectuent l'opération d'ajout d'un 0 *à gauche du ruban.*

- *pour tout* $\alpha \in \{0,1\}$:

$$\begin{array}{rcll} q + X\alpha & \to & \alpha + Xq & \text{(l)} \\ q + Xh' & \to & q' & \text{(m)} \\ \alpha + Xq' & \to & q' & \text{(n)} \\ h + Xq' & \to & q'' & \text{(o)} \end{array}$$

Ces règles réécrivent tout terme représentant un état d'arrêt en q''.

Nous sommes en fait intéressé par la théorie équationnelle définie par ces règles et nous définissons l'ensemble d'équations défini par $\mathcal{M}$:

Définition 6.19 *L'ensemble* $E(\mathcal{M})$ *d'équations closes est* $\{s = t \mid s \to t \in R(\mathcal{M})\}$.

Exemple. Voici un exemple illustrant le rôle des règles (h)-(k). Supposons que le ruban soit $\ldots 001100 \ldots$, la position courante sur le 1 de gauche, et l'état courant soit q_0. Un terme représentant cette configuration est :

$$h + Xq_0 + X^2 1 + X^3 1 + X^4 h'$$

Si maintenant on suppose qu'il y a une transition $\delta(q_0, 1) = (q_1, 0, L)$. $R(\mathcal{M})$ contient alors la règle :

$$h + Xq_0 + X^2 1 \to h + Xq'_1 + X^2 0 + X^3 I0$$

Donc notre terme se réécrit en

$$h + Xq'_1 + X^2 0 + X^3 I0 + X^3 1 + X^4 h'$$

Maintenant l'état est « primé » donc toutes les transitions sont bloquées. La seule règle pouvant s'appliquer (modulo ACD) est (h) :

$$X^3 I0 + X^3 1 =_{\mathrm{ACD}} X^3(I0 + 1) \underset{(h)}{\longrightarrow} X^3(0 + XI1) =_{\mathrm{ACD}} X^3 0 + X^4 I1$$

Nous avons alors la séquence de réductions :

$$\begin{array}{rl} h + Xq'_1 + X^2 0 + X^3 I0 + X^3 1 + X^4 h' & \xrightarrow[(h)/\mathrm{ACD}]{} \\ h + Xq'_1 + X^2 0 + X^3 0 + X^4 I1 + X^4 h' & \xrightarrow[(i)/\mathrm{ACD}]{} \\ h + Xq'_1 + X^2 0 + X^3 0 + X^4 U1 + X^5 h' & \xrightarrow[(j)/\mathrm{ACD}]{} \\ h + Xq'_1 + X^2 0 + X^3 U0 + X^4 1 + X^5 h' & \xrightarrow[(j)/\mathrm{ACD}]{} \\ h + Xq'_1 + X^2 U0 + X^3 0 + X^4 1 + X^5 h' & \xrightarrow[(k)/\mathrm{ACD}]{} \\ h + Xq_1 + X^2 0 + X^3 0 + X^4 1 + X^5 h' & \end{array}$$

et nous constatons (ô miracle !) que chaque exposant a été incrémenté. Maintenant la règle correspondant à la transition $\delta(q_1, 0)$ peut s'appliquer.

6.4.3 Démonstration de la correction de la simulation

Nous allons maintenant montrer que la réécriture par le système $R(\mathcal{M})$ modulo ACD simule exactement le calcul de la machine $\mathcal{M}$. Le point fondamental est de montrer que la réécriture $\xrightarrow[R(\mathcal{M})/ACD]{}$ est déterministe. Cela n'est pas trivial car il y a des superpositions entre les règles de $R(\mathcal{M})$, contrairement au cas A.

Nous commençons par quelques lemmes préliminaires.

Lemme 6.20 *Si deux termes sont égaux modulo $E(\mathcal{M}) \cup$ ACD alors ils ont le même nombre de caractères appartenant à $\tilde{Q} = \{q_0, \ldots, q_N, q'_0, \ldots, q'_N, q, q'\}$.*

Démonstration. Toutes les équations laissent fixe le nombre d'éléments de $\tilde{Q}$. □

Dans ce qui va suivre, nous nous intéressons uniquement aux termes qui ont un sens, c'est-à-dire à ceux qui représentent une configuration valide. Tout d'abord, nous considérons les termes u qui ont exactement un symbole dans $\tilde{Q}$. Ce symbole sera appelé *l'état* de u. Nous définissons alors précisément l'ensemble des termes valides.

Définition 6.21 *L'ensemble des* termes valides *est la réunion des six ensembles suivants. Le premier est celui qui contient les termes représentant une configuration de la machine. Les cinq autres représentent les configurations transitoires quand est effectuée l'opération d'ajout d'un 0 à gauche du ruban, et également de la réécriture vers q'' de tout terme représentant un état d'arrêt :*

- *configurations normales, l'état de u est dans $\{q_0, \ldots, q_N\}$ et u ne contient ni I ni U :*

$$\begin{aligned} S_1 \;=\; & \{h + X\alpha_1 + \ldots \\ & + X^{i-1}\alpha_{i-1} + X^i q_j + X^{i+1}\alpha_{i+1} + \ldots + X^{n+1}\alpha_n + X^{n+2}h'\}; \end{aligned}$$

- *ajout d'un 0 à gauche, première étape, l'état de u est dans $\{q'_0, \ldots, q'_N\}$ et u contient un I et pas de U :*

$$\begin{aligned} S_2 \;=\; & \{h + Xq'_j + X^2\alpha_1 + \ldots \\ & + X^i\alpha_{i-1} + X^{i+1}I\alpha_i + X^{i+1}\alpha_{i+1} + \ldots + X^n\alpha_n + X^{n+1}h'\}; \end{aligned}$$

- *ajout d'un 0 à gauche, deuxième étape, l'état de u est dans $\{q'_0, \ldots, q'_N\}$ et u contient un U et pas de I :*

$$\begin{aligned} S_3 \;=\; & \{h + Xq'_j + X^2\alpha_1 + \ldots \\ & + X^i\alpha_{i-1} + X^{i+1}U\alpha_i + X^{i+2}\alpha_{i+1} + \ldots + X^{n+1}\alpha_n + X^{n+2}h'\}; \end{aligned}$$

- *la machine s'arrête, et pour commencer on déplace la position courante complètement à droite, l'état de u est q et u ne contient ni I ni U :*

$$\begin{aligned} S_4 \;=\; & \{h + X\alpha_1 + \ldots \\ & + X^{i-1}\alpha_{i-1} + X^i q + X^{i+1}\alpha_{i+1} + \ldots + X^{n+1}\alpha_n + X^{n+2}h'\}; \end{aligned}$$

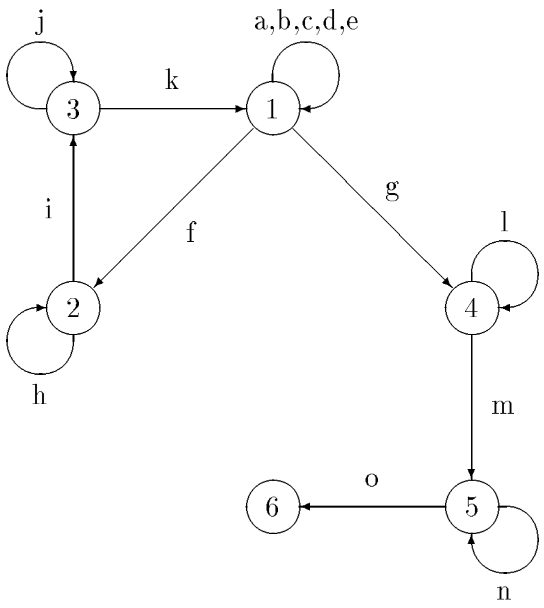

Figure 6.1 : Sources et cibles des règles de $R(\mathcal{M})$

- *effacement de tous les* 0 *et les* 1 *du ruban, l'état de* u *est* q' *et* u *ne contient ni* I *ni* U *:*

$$S_5 = \{h + X\alpha_1 + \ldots + X^n\alpha_n + X^{n+1}q'\};$$

- *terme final, représentant la configuration finale :*

$$S_6 = \{q''\}$$

Les S_i sont trivialement disjoints en regardant l'état et la présence de I ou U. Nous montrons maintenant que chaque règle de $R(\mathcal{M})$ ne peut être appliquée qu'à des termes qui sont dans un S_i particulier, et produit des termes qui sont dans un S_j particulier, éventuellement le même. Nous appelons ces ensembles respectivement la source et la cible de la règle considérée.

Définition 6.22 *Pour chaque règle* (r) *dans* $R(\mathcal{M})$*, la source et la cible de cette règle, notées respectivement* $\mathrm{source}(r)$ *et* $\mathrm{cible}(r)$*, sont définies comme l'un des* S_i*, comme indiqué sur la figure 6.1.*

Par exemple, $\mathrm{source}(g) = S_1$ et $\mathrm{cible}(g) = S_4$. Le lemme suivant montre que les règles de $R(\mathcal{M})$ produisent toujours un terme valide à partir d'un terme valide, et réciproquement ne peuvent donner un terme valide qu'à partir d'un terme valide.

Lemme 6.23 *Si* $s \xrightarrow[(r)/\mathrm{ACD}]{} t$ *pour des termes arbitraires* s *et* t *et pour une règle* (r) *dans* $R(\mathcal{M})$*. Les propriétés suivantes sont équivalentes :*

(i) s *est valide.*

(ii) $s \in \mathrm{source}(r)$

(iii) $t \in \mathrm{cible}(r)$.

(iv) t est valide.

Démonstration. les implications (ii)$\Longrightarrow$(i) et (iii)$\Longrightarrow$(iv) sont triviales. Les implications inverses et (ii)$\Longleftrightarrow$(iii) sont des vérifications évidentes. □

Lemme 6.24 *Soit u un terme valide. Alors il y a au plus une règle pouvant être appliquée à u, et à une seule position dans u.*

Démonstration. par le lemme 6.23 et parce qu'il n'y a pas de superpositions entre des règles qui ont la même source. □

Nous obtenons alors le théorème suivant qui montre que $\xrightarrow[R(\mathcal{M})/\mathrm{ACD}]{}$ simule bien $\mathcal{M}$.

Théorème 6.25 $\langle T, p, q\rangle \overset{*}{\underset{\mathcal{M}}{\Rightarrow}} \langle T', p', q'\rangle$ *si et seulement si*

$$\overline{\langle T, p, q\rangle} \xrightarrow[R(\mathcal{M})/\mathrm{ACD}]{*} \overline{\langle T', p', q'\rangle}$$

Corollaire 6.26 *La machine de Turing $\mathcal{M}$ s'arrête pour la donnée $\alpha_1 \ldots \alpha_n$ si et seulement si :*

$$h + Xq_0 + X^2\alpha_1 + \ldots + X^{n+1}\alpha_n + X^{n+2}h' \xrightarrow[R(\mathcal{M})/\mathrm{ACD}]{*} q''$$

Démonstration. Évidente par le lemme 6.24. □

6.4.4 Réduction du problème de l'arrêt en un problème du mot ACD-clos

Nous avons montré dans la sous-section précédente que le problème de l'arrêt de $\mathcal{M}$ se réduisait en un problème de *réductibilité* pour $R(\mathcal{M})/\mathrm{ACD}$. Mais en général, le problème de réductibilité pour un système de réécriture R n'est pas équivalent au problème du mot de sa théorie équationnelle. Le théorème suivant montre que cette équivalence a lieu pour $R(\mathcal{M})$ si seuls des termes valides sont considérés.

Théorème 6.27 *Pour une donnée $\mathcal{M}$ et n'importe quelle donnée $\alpha_1 \ldots \alpha_n$ on a*

$$h + Xq_0 + X^2\alpha_1 + \ldots + X^{n+1}\alpha_n + X^{n+2}h' \xrightarrow[R(\mathcal{M})/\mathrm{ACD}]{*} q''$$

si et seulement si

$$h + Xq_0 + X^2\alpha_1 + \ldots + X^{n+1}\alpha_n + X^{n+2}h' \xleftrightarrow[E(\mathcal{M})\cup\mathrm{ACD}]{*} q''$$

Démonstration. Le sens direct est trivial. Pour la réciproque, considérons une preuve

$$h + Xq_0 + X^2\alpha_1 + \ldots + X^{n+1}\alpha_n + X^{n+2}h' = u_1 \xleftrightarrow[R(\mathcal{M})]{} u_2 \xleftrightarrow[R(\mathcal{M})]{} \ldots \xleftrightarrow[R(\mathcal{M})]{} u_n = q''$$

$\xleftrightarrow[R(\mathcal{M})]{}$ signifie que la réécriture peut avoir lieu aussi bien à gauche qu'à droite.

Il est évident par le lemme 6.23 que pour chaque étape élémentaire $u_i \underset{(r)}{\longrightarrow} u_{i+1}$, $u_i \in$ source(r) et $u_{i+1} \in$ cible(r); et pour chaque $u_i \underset{(r)}{\longleftarrow} u_{i+1}$, $u_i \in$ cible(r) et $u_{i+1} \in$ source(r).

Supposons que cette preuve ait un pic $u_{i-1} \leftarrow u_i \rightarrow u_{i+1}$. Par le lemme 6.24 nous savons qu'une seule règle peut s'appliquer à u_i, et seulement à une position donnée, par conséquent u_{i-1} et u_{i+1} sont identiques. La preuve peut alors être raccourcie en supprimant le pic. En répétant ce processus, nous obtenons une preuve sans pic. Mais q'' est en forme normale donc en fait nous avons une preuve par réécriture de gauche à droite. □

Nous avons alors, par le théorème 6.8 :

Théorème 6.28 *Il existe un ensemble E d'axiomes clos sur un vocabulaire $\mathcal{F} \cup \{*, +\}$ tel que le problème du mot pour $E \cup$ ACD$(*, +)$ est indécidable.*

6.5 Résumé des résultats de décidabilité et d'indécidabilité

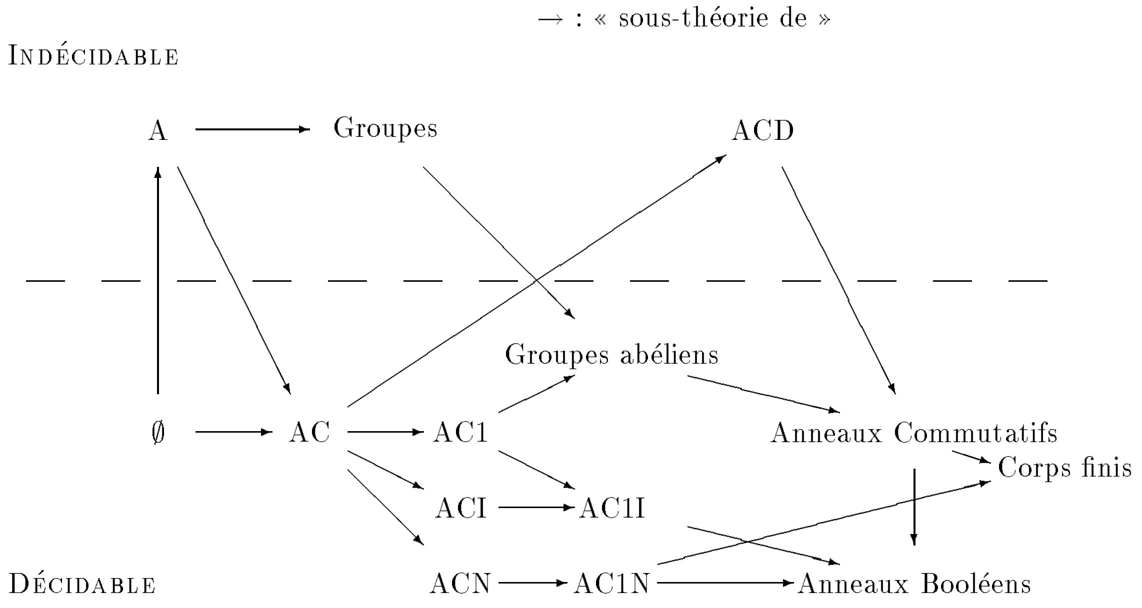

Figure 6.2 : Décidabilité du problème du mot E-clos pour quelques théories E

La figure 6.2 montre les différents résultats de décidabilité ou d'indécidabilité du problème du mot E-clos pour un certain nombre de E.

Conclusion

Grâce à la notion de réécriture normalisée, et l'algorithme de complétion associé, dont nous avons prouvé la complétude en toute généralité, nous avons montré que toute une variété d'algorithmes de « complétion » pouvaient être réunifiés en un seul et même algorithme (figure 6.3).

Ainsi, avec la réécriture normalisée, nous avons pu établir un rapport très étroit entre la complétion de Knuth-Bendix et le calcul de bases standard. Il reste néanmoins encore de nombreux points à éclaircir. Du point du vue théorique, nous n'avons pas réussi à plonger le

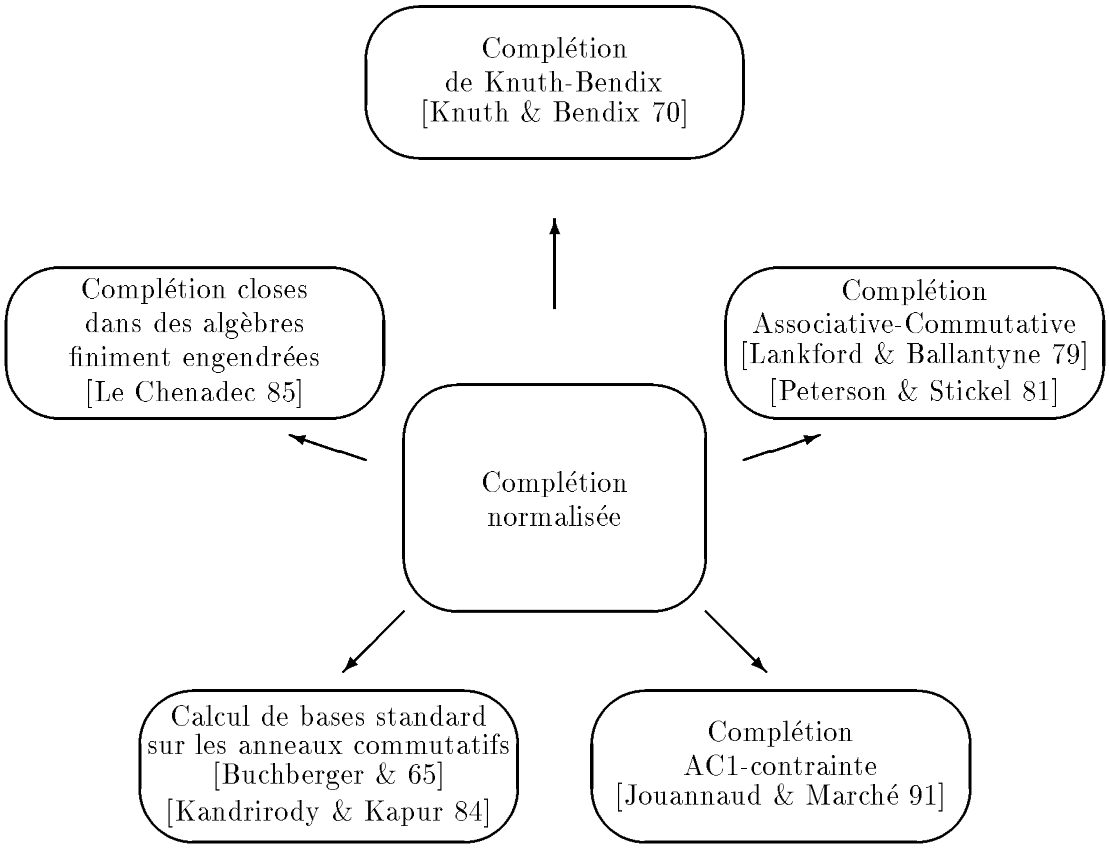

Figure 6.3 : Différents algorithmes connus qui sont instances de la complétion normalisée

calcul de bases standard sur $\mathbb{Q}$ (sur les corps en général) dans la complétion normalisée, à cause de la présence de l'axiome non équationnel $x.x^{-1} \to 1 \mid x \neq 0$. La question qui reste ouverte est existe-il un système convergent S tel que la complétion S-normalisée donne un algorithme de calcul de bases standard sur $\mathbb{Q}$ (plus précisément tel que le résultat de la complétion puisse être traduit trivialement en une base standard, comme dans le cas de $\mathbb{Z}$).

Une approche possible serait peut-être d'étendre les résultats à de la réécriture conditionnelle. Dans le même ordre d'idée, il serait également intéressant de voir s'il est possible d'appliquer les autres extensions connues de la complétion : complétion ordonnée, complétion contrainte, complétion basique, etc.

Du point de vue pratique, il reste aussi des travaux à faire. Il nous faut encore montrer l'efficacité en pratique de l'utilisation de l'unification AC1 ou ACI plutôt que l'unification AC, ainsi que des autres optimisations envisagées : critères d'élimination de paires critiques, compilation du filtrage ou de la normalisation par S, etc.

Annexe A

Implantation de la complétion normalisée

Cette implantation à été faite en Caml-Light, un langage de la famille ML conçu à l'INRIA Rocquencourt [63], avec appel d'une fonction d'unification AC écrite par Boudet [17], en langage C.

A.1 Utilisation du programme

On utilise le compilateur Caml-Light pour fabriquer un programme exécutable `normcomp`, que l'on lance ensuite avec un certain nombre de paramètres :

```
Usage: normcomp [options] <source file name>
options are:
  -s  show states of completion
  -v  verbose: show which inference rule is applied
  -o  orient only, left to right
```

Le fichier source doit contenir une « spécification » dont la syntaxe est donnée par la grammaire de la figure A.1. Cette spécification est constituée de cinq parties.

La première permet de déclarer des noms d'opérateurs, avec soit une arité fixe, soit déclarés associatifs-commutatifs.

La deuxième partie permet de déclarer la théorie S modulo laquelle on va travailler, c'est un ensemble de théories prédéfinies parmi pour l'instant AC1, ACI, AC1I, AC1N (seulement pour la 2-idempotence et la 2-nilpotence), AG, CR et FP. Il faut que les théories données soit disjointes, c'est-à-dire ne partage pas de symboles, et le programme calculera les couples normalisants par réunion des couples pour chacune de ces théories, comme indiqué par le théorème 4.12.

Dans la troisième partie, on donne le système d'équations que l'on veut compléter.

La quatrième partie permet de donner l'ordre sur les termes que l'on va utiliser pour orienter les équations. Il n'y a pour l'instant que deux choix. Premièrement un *rpo*, où il faut donner la précédence (par une liste de chaînes décroissantes d'opérateurs) et éventuellement les statuts (multi-ensemble par défaut). Deuxièmement un ordre « interactif » qui simplement demande à l'utilisateur de comparer à la main, à chaque fois qu'une comparaison est nécessaire. La terminaison est alors bien sûr sous la responsabilité de l'utilisateur !

$$
\begin{array}{rcl}
\mathit{Donnees} & \rightarrow & \texttt{operators}\ \mathit{Operateurs} \\
 & & \texttt{theory}\ \mathit{Theorie} \\
 & & \texttt{axioms}\ \mathit{Equations} \\
 & & \texttt{order}\ \mathit{Ordre} \\
 & & \texttt{end} \\
 & | & \texttt{operators}\ \mathit{Operateurs} \\
 & & \texttt{theory}\ \mathit{Theorie} \\
 & & \texttt{axioms}\ \mathit{Equations} \\
 & & \texttt{order}\ \mathit{Ordre} \\
 & & \texttt{problemes}\ \mathit{Problemes} \\
 & & \texttt{end} \\
\mathit{Operateurs} & \rightarrow & \mathit{DeclArite} \mid \mathit{DeclArite}\ \mathit{Operateurs} \\
\mathit{DeclArite} & \rightarrow & \mathit{OpList}\ \texttt{:}\ \mathit{Arite} \\
\mathit{Arite} & \rightarrow & \texttt{AC} \mid \texttt{constant} \mid \texttt{unary} \mid \texttt{binary} \mid \mathit{Entier} \\
\mathit{OpList} & \rightarrow & \mathit{Ident} \mid \mathit{Ident}\ \texttt{,}\ \mathit{OpList} \\
\mathit{Theorie} & \rightarrow & \mathit{DeclTheorie} \mid \mathit{DeclTheorie}\ \mathit{Theorie} \mid \texttt{empty} \\
\mathit{DeclTheorie} & \rightarrow & \texttt{AC1(}\ \mathit{Ident}\ \texttt{,}\ \mathit{Ident}\ \texttt{)} \\
 & | & \texttt{ACI(}\ \mathit{Ident}\ \texttt{)} \\
 & | & \texttt{AC1I(}\ \mathit{Ident}\ \texttt{,}\ \mathit{Ident}\ \texttt{)} \\
 & | & \texttt{AC1N(}\ \mathit{Ident}\ \texttt{,}\ \mathit{Ident}\ \texttt{)} \\
 & | & \texttt{AG(}\ \mathit{Ident}\ \texttt{,}\ \mathit{Ident}\ \texttt{,}\ \mathit{Ident}\ \texttt{)} \\
 & | & \texttt{CR(}\ \mathit{Ident}\ \texttt{,}\ \mathit{Ident}\ \texttt{,}\ \mathit{Ident}\ \texttt{,}\ \mathit{Ident}\ \texttt{,}\ \mathit{Ident}\ \texttt{)} \\
 & | & \texttt{FP(}\ \mathit{Entier}\ \texttt{,}\ \mathit{Ident}\ \texttt{,}\ \mathit{Ident}\ \texttt{,}\ \mathit{Ident}\ \texttt{,}\ \mathit{Ident}\ \texttt{,}\ \mathit{Ident}\ \texttt{)} \\
\mathit{Equations} & \rightarrow & \mathit{Equation}\ \texttt{;} \mid \mathit{Equation}\ \texttt{;}\ \mathit{Equations} \\
\mathit{Equation} & \rightarrow & \mathit{Term}\ \texttt{=}\ \mathit{Term} \\
\mathit{Ordre} & \rightarrow & \texttt{rpo}\ \mathit{Precedence}\ \texttt{;}\ \mathit{Statuts} \\
 & | & \texttt{rpo}\ \mathit{Precedence} \\
 & | & \texttt{interactive} \\
\mathit{Precedence} & \rightarrow & \mathit{Listeordonnee} \mid \mathit{Listeordonnee}\ \texttt{,}\ \mathit{Precedence} \\
\mathit{Listeordonnee} & \rightarrow & \mathit{Ident} \mid \mathit{Ident}\ \texttt{>}\ \mathit{Listeordonnee} \\
\mathit{Statuts} & \rightarrow & \mathit{Statut} \mid \mathit{Statut}\ \texttt{,}\ \mathit{Statuts} \\
\mathit{Statut} & \rightarrow & \mathit{Ident}\ \texttt{mul} \mid \mathit{Ident}\ \texttt{lrlex} \mid \mathit{Ident}\ \texttt{rllex} \\
\mathit{Problemes} & \rightarrow & \mathit{Probleme}\ \texttt{;} \mid \mathit{Probleme}\ \texttt{;}\ \mathit{Problemes} \\
\mathit{Probleme} & \rightarrow & \mathit{Equation} \mid \texttt{reduce}\ \mathit{Term} \\
\mathit{Term} & \rightarrow & \mathit{Ident}\ \texttt{(}\ \mathit{EndTermList} \mid \mathit{Ident} \mid \mathit{Term}\ \mathit{Ident}\ \mathit{Term} \mid \texttt{(}\ \mathit{Term}\ \texttt{)} \\
\mathit{EndTermList} & \rightarrow & \mathit{Term}\ \texttt{)} \mid \mathit{Term}\ \texttt{,}\ \mathit{EndTermList} \\
\mathit{Ident} & \rightarrow & [\texttt{a}-\texttt{z}, \texttt{A}-\texttt{Z}, \texttt{0}-\texttt{9}, \texttt{\_}]^{+} \mid \texttt{+} \mid \texttt{.} \mid \texttt{\&} \mid \texttt{*} \mid \texttt{-} \mid \texttt{-} \\
\mathit{Entier} & \rightarrow & [\texttt{0}-\texttt{9}]^{+}
\end{array}
$$

Figure A.1 : Grammaire des fichiers sources

Enfin la cinquième partie, qui est optionnelle, permet de donner une liste de « problèmes » à résoudre, qui peuvent être de deux types : soit une équation à prouver ou infirmer, soit tout simplement un terme à normaliser.

Optimisations possibles non implantées :

- permettre les symboles commutatifs mais non associatifs
- autres théories prédéfinies;
- filtrage AC étendu en une seule fonction, pour l'efficacité de la réécriture; optimisation du filtrage : compilation du filtrage pour S; compilation de la normalisation par S;
- unification AC1, AC1I.
- méthodes pour diminuer le nombre de paires critiques : symétrie, blocage [6], superpositions premières [50], etc.

A.2 La stratégie de complétion

Afin de pouvoir exprimer la stratégie utilisée dans l'application des règles d'inférences, on découpe en plusieurs parties l'ensemble des règles et l'ensemble des équations, et l'on définit le contrôle par un ensemble de transitions sur le n-uplet ainsi obtenu. Cette idée est due à Lescanne dans son implantation de la complétion AC nommée Orme [64].

Dans notre implantation, nous utilisons un quintuplet $(R_1, R_2, R_3, E_1, E_2)$. E_1 est l'ensemble des équations que l'on n'a pas encore tenté d'orienter. E_2 est l'ensemble des équations inorientables rencontrées, la seule façcon de les éliminer est de les simplifier.

$R_1 \cup R_2$ est un ensemble de règles dont toutes les superpositions ont été calculées (c'est-à-dire mises dans E_1). R_3 sert à stocker les règles avec lesquelles on n'a pas encore calculé de paires critiques. R_1 sert de stockage intermédiaire lorsqu'on calcule les paires critiques entre une règle de R_3 et toutes les règles de $R_1 \cup R_2$: au départ, R_1 est vide, puis à chaque fois que l'on calcule les paires critiques entre une règle de R_2 la règle de R_3 voulue, cette règle de R_2 est déplacée dans R_1, et enfin lorsque R_2 est devenu vide, on calcule les paires critiques de la règle de R_3 sur elle-même, et on remet le tout dans R_2. Cette technique permet d'assurer l'équité, car à chaque fois que $R_1 = R_3 = \emptyset$, on a $CP_T(R_2) \subseteq E_1 \cup E_2$.

Les règles de transitions sont celles de la figure A.2. Les règles de simplification sont toujours appliquées de façon prioritaire, et les nouvelles équations obtenues sont mises dans E_1, et les nouvelles règles dans R_3.

En spécifiant l'option `-s`, le programme affiche les états successifs de la dérivation.

Il faut remarquer que les règles de transitions ne sont pas déterministes : on peut choisir de faire une orientation alors même que l'on n'a pas fini de calculer les paires critiques entre une règle de R_3 donnée et les règles de $R_1 \cup R_2$. Cela permet d'arrêter de déduire lorsque l'on a engendré une équation « intéressante », c'est-à-dire en fait une équation susceptible de donner une règle qui permettrait de simplifier. On peut alors préférer d'orienter plutôt que de continuer à déduire. Dans l'implantation présente, ce choix est effectué uniquement en fonction de la taille : si l'on a engendré une équation « petite », on essaie de l'orienter. Dans le même ordre d'idée, lorsqu'on doit orienter, il vaut mieux choisir judicieusement l'équation de E_1 à orienter, et dans l'implantation actuelle on choisit l'équation de plus petite taille. En fait, pour ces deux critères il vaudrait probablement mieux utiliser non pas l'ordre sur les

Orientation

$$(R_1, R_2, R_3, \{u = v\} \cup E_1, E_2) \vdash (\emptyset, R_1 \cup R_2, R_3 \cup \Psi(u,v), E_1 \cup \Theta(u,v), E_2)$$
si $u \succ v$

Non-orientabilité

$$(R_1, R_2, R_3, \{u = v\} \cup E_1, E_2) \vdash (R_1, R_2, R_3, E_1, \{u = v\} \cup E_2)$$
si u et v incomparables

Déduction entre deux règles distinctes

$$(R_1, \{r_1\} \cup R_2, \{r_2\} \cup R_3, E_1, E_2) \vdash (\{r_1\} \cup R_1, R_2, \{r_2\} \cup R_3, E_1 \cup CP_T(r_1, r_2), E_2)$$

Déduction d'une règle sur elle-même

$$(R_1, \emptyset, \{r\} \cup R_3, E_1, E_2) \vdash (\emptyset, \{r\} \cup R_1, R_3, E_1 \cup CP_T(r, r), E_2)$$

État initial

$$(\emptyset, \emptyset, \emptyset, E_0, \emptyset)$$
où E_0 est l'ensemble initial d'équations

État de succès

$$(\emptyset, R_2, \emptyset, \emptyset, \emptyset)$$
et l'ensemble R_2 est alors le système canonique cherché

État d'échec

$$(\emptyset, R_2, \emptyset, \emptyset, E_2)$$
si E_2 non vide

Figure A.2 : Règles de transitions définissant la stratégie de complétion

tailles, mais plutôt l'ordre de spécialisation : par exemple, entre les deux équations de même taille $0 * 0 = 0$ et $x * 0 = 0$, il vaudrait mieux choisir la deuxième, qui simplifie la première, et non pas la première comme cela se passe dans l'implantation actuelle pendant la complétion des anneaux.

A.3 Exemples

A.3.1 Un problème de mot : le problème des caméléons

Ce problème permet déjà de voir le principe de fonctionnement du programme. Le fichier source est le suivant

```
operators
  & : AC
  Rouge,Vert,Bleu : constant

theory
  empty

axioms
  Rouge & Bleu = Vert & Vert;
  Rouge & Vert = Bleu & Bleu;
  Vert & Bleu  = Rouge & Rouge;

order
  rpo Rouge>Bleu>Vert > & ; & mul

problems
  Rouge & Rouge & Rouge & Rouge & Bleu & Bleu & Bleu & Vert & Vert =
  Rouge & Rouge & Rouge & Rouge & Rouge & Rouge & Rouge & Rouge & Rouge;
  Rouge & Rouge & Rouge & Rouge & Bleu & Bleu & Bleu & Vert & Vert =
  Bleu & Bleu & Bleu & Bleu & Bleu & Bleu & Bleu & Bleu & Bleu;
  Rouge & Rouge & Rouge & Rouge & Bleu & Bleu & Bleu & Vert & Vert =
  Vert & Vert & Vert & Vert & Vert & Vert & Vert & Vert & Vert;

end
```

On déclare le symbole `&` comme symbole AC et les trois couleurs comme constantes. La théorie S est ici vide, et les trois axiomes sont les trois changements de couleurs possibles. L'ordre un *rpo* qui donne un ordre AC, et enfin l'on donne à prouver les trois équations exprimant qu'une configuration unicolore est atteignable à partir de la configuration initiale composée de quatre caméléons rouges, trois bleus et deux verts. Voici le script de la complétion obtenue :

```
 ----------------------------------------------------
|  Completion normalisee - version 0.5 - 30 mar 93   |
 ----------------------------------------------------

Lecture de cameleon.nc...
----------------------
The'orie : Vide
----------------------
Axiomes :
{Rouge & Bleu = Vert & Vert,
Rouge & Vert = Bleu & Bleu,
Vert & Bleu = Rouge & Rouge} (3 equations)
----------------------
Ordre : RPO
pre'ce'dence :
Rouge > Bleu > Vert > &
statuts :
```

```
& multiset

-----------------------------------------
Comple'tion normalise'e modulo la the'orie
Vide
C'est-a-dire les regles
{}
de l'ensemble d'equations :
{Rouge & Bleu = Vert & Vert,
Rouge & Vert = Bleu & Bleu,
Vert & Bleu = Rouge & Rouge} (3 equations)
-----------------------------------------
Orientation de l'e'quation Vert & Bleu = Rouge & Rouge
en
{Rouge & Rouge -> Vert & Bleu} (1 regles)
Avec les equations critiques :
{}
Orientation de l'e'quation Rouge & Vert = Bleu & Bleu
en
{Rouge & Vert -> Bleu & Bleu} (1 regles)
Avec les equations critiques :
{}
Orientation de l'e'quation Rouge & Bleu = Vert & Vert
en
{Rouge & Bleu -> Vert & Vert} (1 regles)
Avec les equations critiques :
{}
Orientation de l'e'quation Bleu & Bleu & Bleu = Vert & Vert & Vert
en
{Bleu & Bleu & Bleu -> Vert & Vert & Vert} (1 regles)
Avec les equations critiques :
{}

Resultat :
{Rouge & Bleu -> Vert & Vert,
Rouge & Vert -> Bleu & Bleu,
Rouge & Rouge -> Vert & Bleu,
Bleu & Bleu & Bleu -> Vert & Vert & Vert} (4 regles)

Nombre d'appels au filtrage AC     : 420
Nombre d'appels reussis            : 15 (3%)
Nombre d'appels a l'unification AC : 36
Nombre de solutions trouvees       : 20 (0 en moyenne)

Re'solution des proble`mes :

Rouge & Rouge & Rouge & Rouge & Bleu & Bleu & Bleu & Vert & Vert =
Rouge & Rouge & Rouge & Rouge & Rouge & Rouge & Rouge & Rouge & Rouge est fausse
(formes normales Vert & Vert & Vert & Vert & Vert & Vert & Vert & Bleu & Bleu
et Vert & Vert & Vert & Vert & Vert & Vert & Vert & Vert & Vert)

Rouge & Rouge & Rouge & Rouge & Bleu & Bleu & Bleu & Vert & Vert =
Bleu & Bleu & Bleu & Bleu & Bleu & Bleu & Bleu & Bleu & Bleu est fausse
(formes normales Vert & Vert & Vert & Vert & Vert & Vert & Vert & Bleu & Bleu
et Vert & Vert & Vert & Vert & Vert & Vert & Vert & Vert & Vert)

Rouge & Rouge & Rouge & Rouge & Bleu & Bleu & Bleu & Vert & Vert =
```

```
Vert & Vert & Vert & Vert & Vert & Vert & Vert & Vert & Vert est fausse
(formes normales Vert & Vert & Vert & Vert & Vert & Vert & Vert & Bleu & Bleu
et Vert & Vert & Vert & Vert & Vert & Vert & Vert & Vert & Vert)

Temps utilisateur : 1.390 s
Temps syste`me : 0.130 s
```

La complétion a calculé le système canonique et a montré que les trois équations sont invalides, ce qui répond donc négativement au problème des caméléons : on ne peut pas atteindre de configuration stable.

A.3.2 Utilité de NON-ORIENTABILITÉ dans la stratégie : complétion des anneaux commutatifs

Durant la complétion, il est possible de rencontrer une équation inorientable, et pourtant il serait dommage de tomber immédiatement en échec. En effet, on peut « retarder » l'orientation de cette équation, chercher à en orienter une autre, qui pourrait alors produire une règle permettant de réduire l'équation inorientable qui nous gêne, supprimant ainsi le cas d'échec. Ce phénomène est illustré sur l'exemple très classique de la complétion de la théorie des anneaux commutatifs, où l'on rencontre l'équation inorientable $x * (-y) = (-x) * y$ avant d'essayer d'orienter la « bonne » équation $x * (-y) = -(x * y)$.

```
 ----------------------------------------------------
|  Completion normalisee - version 0.6 - 04 mai 93   |
 ----------------------------------------------------

Lecture de anneaux.nc...
----------------------
Theorie : Vide
----------------------
Axiomes :
{x+0 = x,
-(x)+x = 0,
x.1 = x,
x.(y+z) = (x.y)+(x.z)} (4 equations)
----------------------
Ordre : RPO
pre'ce'dence :
. > 1
. > - > 0 > +
statuts :
+ multiset
. multiset

------------------------------------------
Completion normalisee modulo la theorie
Vide
C'est-a-dire les regles
{}
de l'ensemble d'equations :
{x+0 = x,
-(x)+x = 0,
x.1 = x,
x.(y+z) = (x.y)+(x.z)} (4 equations)
------------------------------------------
```

```
...
 ---- ORIENTATION de -(x).y = -(y).x
Equation inorientable, postpone
 ---- ORIENTATION de y.-(x) = -(x.y)
Orientation de l'equation y.-(x) = -(x.y)
en
{y.-(x) -> -(x.y)} (1 regles)
Avec les equations critiques :
{}
...
Resultat :
{-(0) -> 0,
x.0 -> 0,
x+0 -> x,
-(-(x)) -> x,
x.1 -> x,
-(x)+x -> 0,
-(y+x) -> -(x)+-(y),
y.-(x) -> -(x.y),
x.(y+z) -> (x.y)+(x.z)} (9 regles)

Nombre d'appels au filtrage AC     : 23925
Nombre d'appels reussis            : 1982 (8%)
Nombre d'appels a l'unification AC : 136
Nombre de solutions trouvees       : 537 (3.95 en moyenne)
Re'solution des proble`mes :

(x+y).(x+y) = (x.x)+(x.y)+(x.y)+(y.y) est vraie
(forme normale commune (y.y)+(y.x)+(x.y)+(x.x))

(x+-(y)).(x+-(y)) = (x.x)+-(x.y)+-(x.y)+(y.y) est vraie
(forme normale commune (y.y)+-(y.x)+-(y.x)+(x.x))

(x+y).(x+-(y)) = (x.x)+-(y.y) est vraie
(forme normale commune -(y.y)+(x.x))

Temps utilisateur : 59.960 s
Temps systeme : 3.480 s
```

Bibliographie

[1] W. Ackermann. *Solvable Cases of the Decision Problem.* North-Holland, Amsterdam, 1954.

[2] Siva Anantharaman et Jieh Hsiang. Automated proofs of the moufang identities in alternative rings. *Journal of Automated Reasoning*, 6:79–109, 1990.

[3] Siva Anantharaman, Jieh Hsiang et Jalel Mzali. SbReve2: a term rewriting laboratory with (AC)-unfailing completion. *Proc. 3rd Rewriting Techniques and Applications, Chapel Hill, LNCS 355.* Springer-Verlag, 1989.

[4] Leo Bachmair. Associative-commutative reduction orderings. *Info Proc. Letters*, 1992.

[5] Leo Bachmair et Nachum Dershowitz. Commutation, transformation, and termination. Jorg H. Siekmann, éditeur, *Proc. 8th Int. Conf. on Automated Deduction, Oxford, England, LNCS 230*, pages 5–20, Juillet 1986.

[6] Leo Bachmair et Nachum Dershowitz. Critical pair criteria for completion. *Journal of Symbolic Computation*, 6(1):1–18, 1988.

[7] Leo Bachmair et Nachum Dershowitz. Completion for rewriting modulo a congruence. *Theoretical Computer Science*, 67(2&3):173–201, Octobre 1989.

[8] Leo Bachmair, Nachum Dershowitz et Jieh Hsiang. Orderings for equational proofs. *Proc. 1st IEEE Symp. Logic in Computer Science, Cambridge, Mass.*, pages 346–357, Juin 1986.

[9] Leo Bachmair, Harald Ganzinger, Christopher Lynch et Wayne Snyder. Basic paramodulation and superposition. Deepak Kapur, éditeur, *Proc. 11th Int. Conf. on Automated Deduction, Saratoga Springs, NY, LNCS 607.* Springer-Verlag, Juin 1992.

[10] Leo Bachmair et David A. Plaisted. Termination orderings for associative-commutative rewriting systems. *Journal of Symbolic Computation*, 1(4):329–349, Décembre 1985.

[11] Timothy Baird, Gerald Peterson et Ralph Wilkerson. Complete sets of reductions modulo Associativity, Commutativity and Identity. *Proc. 3rd Rewriting Techniques and Applications, Chapel Hill, LNCS 355*, pages 29–44. Springer-Verlag, Avril 1989.

[12] Jon Barwise, éditeur. *Handbook of Mathematical Logic*, chapitre Recursion Theory. North-Holland, 1977.

[13] Ahlem Ben Cherifa et Pierre Lescanne. An actual implementation of a procedure that mechanically proves termination of rewriting systems based on inequalities between polynomial interpretations. *Proc. 8th Int. Conf. on Automated Deduction, Oxford, England, LNCS 230*, pages 42–51. Springer-Verlag, Juillet 1986.

[14] W. Boone. The word problem. *Annals of Math.*, 70:207–265, 1959.

[15] E. Börger. *Computability, Complexity, Logic*, volume 128 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, 1990.

[16] Alexandre Boudet. Unification dans les mélanges de théories équationnelles. Thèse de Doctorat, Université de Paris-Sud, France, Février 1990.

[17] Alexandre Boudet. Competing for the AC-unification race. *Journal of Automated Reasoning*, 11:185–212, 1993.

[18] Alexandre Boudet, Jean-Pierre Jouannaud et Manfred Schmidt-Schauss. Unification in Boolean rings and Abelian groups. *Journal of Symbolic Computation*, 8:449–477, 1989.

[19] Bruno Buchberger. *An Algorithm for Finding a Basis for the Residue Class Ring of a Zero-Dimensional Ideal.* PhD thesis, University of Innsbruck, Austria, 1965. (in German).

[20] Bruno Buchberger. History and basic features of the critical pair / completion procedure. *Journal of Symbolic Computation*, 3(1), Février 1987.

[21] Reinhard Bündgen. Simulating Buchberger's algorithm by a Knuth-Bendix completion procedure. Ronald V. Book, éditeur, *Proc. 4th Rewriting Techniques and Applications, LNCS 488*, Como, Italy, Avril 1991. Springer-Verlag.

[22] Hans-Jurgen Bürckert, Alexander Herold, Deepak Kapur, Jorg H. Siekmann, Mark E. Stickel, Michael Tepp et Hantao Zhang. Opening the AC-unification race. *Journal of Automated Reasoning*, 4(4):465–474, Décembre 1988.

[23] Martin Davis. Unsolvable problems. J. Barwise, éditeur, *Handbook of Mathematical Logic*, pages 567–594. North-Holland, 1977.

[24] Catherine Delor. Terminaison de systèmes de réécriture, application à la transformation de formules équationnelles. Thèse de Doctorat, Université Paris VII, France, 1991.

[25] Catherine Delor et Laurence Puel. Extension of the associative path ordering to a chain of associative-commutative symbols. *Proc. 5th Rewriting Techniques and Applications, Montréal, LNCS 690*, 1993.

[26] Nachum Dershowitz. A note on simplification orderings. *Information Processing Letters*, 9(5):212–215, Novembre 1979.

[27] Nachum Dershowitz. Orderings for term rewriting systems. *Theoretical Computer Science*, 17(3):279–301, Mars 1982.

[28] NACHUM DERSHOWITZ. Termination of rewriting. *Journal of Symbolic Computation*, 3(1):69–115, Février 1987.

[29] NACHUM DERSHOWITZ et JEAN-PIERRE JOUANNAUD. Notations for rewriting. *EATCS Bulletin*, 43:162–172, 1990.

[30] NACHUM DERSHOWITZ et JEAN-PIERRE JOUANNAUD. Rewrite systems. J. VAN LEEUWEN, éditeur, *Handbook of Theoretical Computer Science*, volume B, pages 243–309. North-Holland, 1990.

[31] NACHUM DERSHOWITZ et ZOHAR MANNA. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, Août 1979.

[32] HERVÉ DEVIE. Une approche algébrique de la réécriture de preuves équationnelles et son application à la dérivation de procédures de complétion. Thèse de Doctorat, Université de Paris-Sud, France, Octobre 1991.

[33] ERIC DOMENJOUD. Outils pour la déduction automatique dans les théories associatives-commutatives. Thèse de doctorat de l'université de Nancy I, 1991.

[34] ERIC DOMENJOUD. Number of minimal unifiers of the equation $\alpha x_1 + \cdots + \alpha x_p \doteq_{AC} \beta y_1 + \cdots + \beta y_q$. *Journal of Automated Reasoning*, 1992.

[35] V. A. EMILIČEV. On algorithmic decidability of certain mass problems in the theory of commutative semigroups. *Sbirsk. Mat. Ž*, 4:788–798, 1963. In russian.

[36] T. EVANS. On multiplicative systems defined by generators and relations, I. *Proceedings of the Cambridge Philosophical Society*, 47:637–649, 1951.

[37] ANDRÉ GALLIGO. Algorithmes de calcul de bases standard. Université de Nice, 1983.

[38] JACQUES HERBRAND. Recherches sur la théorie de la démonstration. Thèse d'Etat, Univ. Paris, 1930. Also in: Ecrits logiques de Jacques Herbrand, PUF, Paris, 1968.

[39] GRAHAM HIGMAN. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society*, 2(3):326–336, Septembre 1952.

[40] DAVID HILBERT. Über die theorie der algebrischen formen. *Math. Ann.*, 36:473–534, 1890.

[41] GÉRARD HUET. Confluent reductions: abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, Octobre 1980.

[42] GÉRARD HUET. A complete proof of correctness of the Knuth-Bendix completion algorithm. *Journal of Computer and System Sciences*, 23:11–21, 1981.

[43] JEAN-PIERRE JOUANNAUD et HÉLÈNE KIRCHNER. Completion of a set of rules modulo a set of equations. *SIAM Journal on Computing*, 15(4):1155–1194, 1986.

[44] JEAN-PIERRE JOUANNAUD et HÉLÈNE KIRCHNER. Completion of a set of rules modulo a set of equations. *SIAM Journal on Computing*, 15(4), Novembre 1986.

[45] JEAN-PIERRE JOUANNAUD et CLAUDE MARCHÉ. Termination and completion modulo associativity, commutativity and identity. *Theoretical Computer Science*, 104:29–51, 1992.

[46] JEAN-PIERRE JOUANNAUD et M. MUNOZ. Termination of rules modulo a set of equations. *Proc. 7th Int. Conf. on Automated Deduction, Napa, LNCS 170*, pages 175–193. Springer-Verlag, 1984.

[47] S. KAMIN et JEAN-JACQUES LÉVY. Two generalizations of the recursive path ordering. Available as a report of the department of computer science, University of Illinois at Urbana-Champaign, 1980.

[48] ABDELILAH KANDRI-RODY et DEEPAK KAPUR. An algorithm for computing the Gröbner basis of a polynomial ideal over an Euclidean ring. Rapport technique 84CRD045, CRD, General Electric Company, Schenectady, New-York, Décembre 1984.

[49] ABDELILAH KANDRI-RODY, DEEPAK KAPUR et FRANZ WINKLER. Knuth-Bendix procedure and Buchberger algorithm — a synthesis. *Proc. of the 20th Int. Symp. on Symbolic and Algebraic Computation, Portland, Oregon*, pages 55–67, 1989.

[50] DEEPAK KAPUR, D.R. MUSSER et PALIATH NARENDRAN. Only prime superpositions need be considered for the Knuth-Bendix procedure. *Journal of Symbolic Computation*, 4:19–36, 1988.

[51] DEEPAK KAPUR et PALIATH NARENDRAN. Constructing Gröbner bases for a polynomial ideal. Presented at the workshop *Combinatorial algorithms in algebraic structures* at Europaische Akademie, Otzenhause, 1985.

[52] DEEPAK KAPUR et PALIATH NARENDRAN. A finite Thue system with decidable word problem and without equivalent finite canonical system. *Theoretical Computer Science*, 35:337–344, 1985.

[53] DEEPAK KAPUR et PALIATH NARENDRAN. NP-completeness of set unification and matching problems. *Proc. 8th Int. Conf. on Automated Deduction, Oxford, England, LNCS 230*, pages 489–495. Springer-Verlag, 1986.

[54] CLAUDE KIRCHNER, éditeur. *Unification.* Academic Press, 1990.

[55] DONALD E. KNUTH et PETER B. BENDIX. Simple word problems in universal algebras. J. LEECH, éditeur, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

[56] JOSEPH B. KRUSKAL. *The theory of well-partially-ordered sets.* PhD thesis, Princeton University, Princeton, NJ, Juin 1954.

[57] JOSEPH B. KRUSKAL. Well-quasi-ordering, the Tree Theorem, and Vazsonyi's conjecture. *Transactions of the American Mathematical Society*, 95:210–225, Mai 1960.

[58] YVES LAFONT et ALAIN PROUTÉ. Church-rosser property and homology of monoïds. Rapport de recherche 90-24, Lab. d'Informatique de l'ENS, 45, Rue d'Ulm, 75230 Paris cedex 05, France, Novembre 1990.

[59] Dallas S. Lankford. Canonical algebraic simplification in computational logic. Memo ATP-25, Automatic Theorem Proving Project, Univ. Texas, Austin, 1975.

[60] Dallas S. Lankford. On proving term rewriting systems are noetherian. Rapport technique MTP-3, Mathematics Department, Louisiana Tech. Univ., 1979.

[61] Dallas S. Lankford et A. M. Ballantyne. Decision procedures for simple equational theories with commutative-associative axioms: Complete sets of commutative-associative reductions. Research Report Memo ATP-39, Department of Mathematics and Computer Science, University of Texas, Austin, Texas, USA, Août 1977.

[62] Philippe Le Chenadec. *Canonical forms in finitely presented algebras.* Pitman, London, 1986.

[63] Xavier Leroy et Michel Mauny. *The Caml Light system, release 0.5.* INRIA, Rocquencourt, 1992.

[64] Pierre Lescanne. Implementation of completion by transition rules + control: Orme. *Proceedings of the Second International Conference on Algebraic and Logic Programming, LNCS 463*, Octobre 1990.

[65] A.I. Mal'cev. On homomorphisms of finite groups. *Ivano Gosudarstvenni Pedag. Inst. Ucheneye Zap*, 18:49–90, 1958.

[66] Claude Marché. On ground AC-completion. Ronald V. Book, éditeur, *Proc. 4th Rewriting Techniques and Applications, LNCS 488*, Como, Italy, Avril 1991. Springer-Verlag.

[67] A. A. Markov. On the impossibility of certain algorithms in the theory of associative systems. *Dokl. Akad. Nauk SSSR*, 55(7):587–590, 1947. In Russian, English translation in C.R. Acad. Sci. URSS, 55, 533-586.

[68] J. V. Matijasevic. Simple examples of undecidable associative calculi. *Soviet Mathematics (Dokladi)*, 8(2):555–557, 1967.

[69] Ernst W. Mayr et Albert R. Meyer. The complexity of the word problem for commutative semigroups and polynomial ideals. *Advances in Mathematics*, 46:305–329, 1982.

[70] Chetan R. Murthy et James R. Russell. A constructive proof of Higman's lemma. *Proc. 5th IEEE Symp. Logic in Computer Science, Philadelphia*, pages 257–267, 1990.

[71] Paliath Narendran et Michaël Rusinowitch. Any ground associative-commutative theory has a finite canonical system. Ronald V. Book, éditeur, *Proc. 4th Rewriting Techniques and Applications, LNCS 488*, Como, Italy, Avril 1991. Springer-Verlag.

[72] Paliath Narendran et Michaël Rusinowitch. The unifiability problem in ground AC theories. *Proc. 8th IEEE Symp. Logic in Computer Science, Montréal*, 1993.

[73] C. St. J. A. Nash-Williams. On well-quasi-ordering finite trees. *Proceedings of the Cambridge Philosophical Society*, 59(4):833–835, Octobre 1963.

[74] M. H. A. NEWMAN. On theories with a combinatorial definition of 'equivalence'. *Ann. Math.*, 43(2):223–243, 1942.

[75] ROBERT NIEUWENHUIS et ALBERT RUBIO. Theorem proving with ordering constrained clauses. DEEPAK KAPUR, éditeur, *Proc. 11th Int. Conf. on Automated Deduction, Saratoga Springs, NY, LNCS 607*. Springer-Verlag, Juin 1992.

[76] ROBERT NIEUWENHUIS et ALBERT RUBIO. A precedence-based total AC-compatible ordering. CLAUDE KIRCHNER, éditeur, *Proc. 5th Rewriting Techniques and Applications, Montréal, LNCS 690*. Springer-Verlag, Juin 1993.

[77] P. S. NOVIKOV. On the algorithmic unsolvability of the word problem in group theory. *Trudy Mat. Inst. Steklov*, 44:1–143, 1955. in Russian.

[78] GERALD E. PETERSON et MARK E. STICKEL. Complete sets of reductions for some equational theories. *Journal of the ACM*, 28(2):233–264, Avril 1981.

[79] DAVID A. PLAISTED. A recursively defined ordering for proving termination of term rewriting systems. Unpublished report R-78-943, University of Illinois, Urbana, IL, 1978.

[80] EMIL L. POST. Recursive unsolvability of a problem of Thue. *Journal of Symbolic Logic*, 13:1–11, 1947.

[81] LOÏC POTTIER. Algorithmes de complétion et généralisation en logique du premier ordre. Thèse de doctorat. Université de Nice, 1989.

[82] MICHAËL RUSINOWITCH et LAURENT VIGNERON. Automated deduction with associative-commutative operators. *Fundamentals of Artificial Intelligence, P. Jorrand Editor, LNCS 535*. Springer-Verlag, 1991.

[83] P. RUZICKA et IGOR PRIVARA. An almost linear Robinson unification algorithm. *Acta Informatica*, 27:61–71, 1989.

[84] C. C. SQUIER. Word problems and a homological finiteness condition for monoids. *Journal of Pure and Applied Algebra*, 49:201–217, 1987.

[85] JOHN STILLWELL. The word problem and the isomorphism problem for groups. *Bulletin of the American Mathematical Society*, 6(1):33–56, Janvier 1982.

[86] Y. TOYAMA. On the Church-Rosser property for the direct sum of term rewriting systems. *Journal of the ACM*, 34(1):128–143, Avril 1987.

[87] ALAN M. TURING. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, 42(2):230–265, 1936.

[88] ALAN M. TURING. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, 43:544–546, 1937.

Table des figures

Index