Unification et disunification : théorie et applications

Hubert Comon

THÈSE

présentée par

Hubert COMON

pour obtenir le grade de Docteur
de l'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE
(arrêté ministériel du 5 juillet 1984)

Spécialité : INFORMATIQUE

UNIFICATION ET DISUNIFICATION. THÉORIE ET APPLICATIONS.

Date de soutenance : 18 mars 1988

Composition du jury:

JP. Verjus (président)
J. Calmet
A. Colmerauer
G. Huet
JP. Jouannaud
C. Kirchner
P. Lescanne

Thèse prèparée au sein du Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle (LIFIA).

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Président : Georges LESPINARD

Année 1988

Professeurs des Universités

BARIBAUD Michel	ENSERG
BARRAUD Alain	ENSIEG
BAUDELET Bernard	ENSPG
BEAUFILS Jean-Pierre	ENSEEG
BLIMAN Samuel	ENSERG
BLOCH Daniel	ENSPG
BOIS Philippe	ENSHMG
BONNETAIN Lucien	ENSEEG
BOUVARD Maurice	ENSHMG
BRISSONNEAU Pierre	ENSIEG
BRUNET Yves	IUFA
CAILLERIE Denis	ENSHMG
CAVAIGNAC Jean-François	ENSPG
CHARTIER Germain	ENSPG
CHENEVIER Pierre	ENSERG
CHERADAME Hervé	UFR PGP
CHOVET Alain	ENSERG
COHEN Joseph	ENSERG
COUMES André	ENSERG
DARVE Félix	ENSHMG
DELLA-DORA Jean	ENSIMAG
DEPORTES Jacques	ENSPG
DOLMAZON Jean-Marc	ENSERG
DURAND Francis	ENSEEG
DURAND Jean-Louis	ENSIEG
FOGGIA Albert	ENSIEG
FONLUPTJean	ENSIMAG
FOULARD Claude	ENSIEG
GANDINI Alessandro	UFR PGP
GAUBERT Claude	ENSPG
GENTIL Pierre	ENSERG
GREVEN Hélène	IUFA
GUERIN Bernard	ENSERG
GUYOT Pierre	ENSEEG
IVANES Marcel	ENSIEG
JAUSSAUD Pierre	ENSIEG
JOUBERT Jean-Claude	ENSPG
JOURDAIN Geneviève	ENSIEG
LACOUME Jean-Louis	ENSIEG
LESIEUR Marcel	ENSHMG
LESPINARD Georges	ENSHMG
LONGEQUEUE Jean-Pierre	ENSPG
LOUCHET François	ENSIEG
MASSE Philippe	ENSIEG
MASSELOT Christian	ENSIEG
MAZARE Guy	ENSIMAG
MOREAU René	ENSHMG
MORET Roger	ENSIEG
MOSSIERE Jacques	ENSIMAG
OBLED Charles	ENSHMG
OZIL Patrick	ENSEEG
PARIAUD Jean-Charles	ENSEEG
PERRET René	ENSIEG
PERRET Robert	ENSIEG
PIAU Jean-Michel	ENSHMG
POUPOT Christian	ENSERG
RAMEAU Jean-Jacques	ENSEEG
RENAUD Maurice	UFR PGP
ROBERT André	UFR PGP
ROBERT François	ENSIMAG
SABONNADIERE Jean-Claude	ENSIEG
SAUCIER Gabrielle	ENSIMAG
SCHLENKER Claire	ENSPG
SCHLENKER Michel	ENSPG
SILVY Jacques	UFR PGP
SIRIEYS Pierre	ENSHMG
SOHM Jean-Claude	ENSEEG
SOLER Jean-Louis	ENSIMAG
SOUQUET Jean-Louis	ENSEEG
TROMPETTE Philippe	ENSHMG
VEILLON Gérard	ENSIMAG
ZADWORNY François	ENSERG

Professeur Université des Sciences Sociales (Grenoble II)

BOLLIET Louis

Personnes ayant obtenu le diplôme d'HABILITATION A DIRIGER DES RECHERCHES

BECKER Monique
BINDER Zdenek
CHASSERY Jean-Marc
CHOLLET Jean-Pierre
COEY John
COLINET Catherine
COMMAULT Christian
CORNUEJOLS Gérard
COULOMB Jean- Louis
DALARD Francis
DANES Florin
DEROO Daniel
DIARD Jean-Paul
DION Jean-Michel
DUGARD Luc
DURAND Madeleine
DURAND Robert
GALERIE Alain
GAUTHIER Jean-Paul
GENTIL Sylviane
GHIBAUDO Gérard
HAMAR Sylvaine
HAMAR Roger
LADET Pierre
LATOMBE Claudine
LE GORREC Bernard
MADAR Roland
MULLER Jean
NGUYEN TRONG Bernadette
PASTUREL Alain
PLA Fernand
ROUGER Jean
TCHUENTE Maurice
VINCENT Henri

Chercheurs du C.N.R.S Directeurs de recherche 1ère Classe

CARRE René
FRUCHART Robert
HOPFINGER Emile
JORRAND Philippe
LANDAU Ioan
VACHAUD Georges
VERJUS Jean-Pierre

Directeurs de recherche 2ème Classe

ALEMANY Antoine
ALLIBERT Colette
ALLIBERT Michel
ANSARA Ibrahim
ARMAND Michel
BERNARD Claude
BINDER Gilbert
BONNET Roland
BORNARD Guy
CAILLET Marcel
CALMET Jacques
COURTOIS Bernard
DAVID René
DRIOLE Jean
ESCUDIER Pierre
EUSTATHOPOULOS Nicolas
GUELIN Pierre
JOUD Jean-Charles
KLEITZ Michel
KOFMAN Walter
KAMARINOS Georges
LEJEUNE Gérard
LE PROVOST Christian
MADAR Roland
MERMET Jean
MICHEL Jean-Marie
MUNIER Jacques
PIAU Monique
SENATEUR Jean-Pierre
SIFAKIS Joseph
SIMON Jean-Paul
SUERY Michel
TEODOSIU Christian
VAUCLIN Michel
WACK Bernard

Personnalités agrées à titre permanent à diriger des travaux de recherche (décision du conseil scientifique)

E.N.S.E.E.G

CHATILLON Christian
HAMMOU Abdelkader
MARTIN GARIN Régina
SARRAZIN Pierre
SIMON Jean-Paul

E.N.S.E.R.G

BOREL Joseph

E.N.S.I.E.G

DESCHIZEAUX Pierre
GLANGEAUD François
PERARD Jacques
REINISCH Raymond

E.N.S.H.G

ROWE Alain

E.N.S.I.M.A.G

COURTIN Jacques

E.F.P.

CHARUEL Robert

C.E.N.G

CADET Jean
COEURE Philippe
DELHAYE Jean-Marc
DUPUY Michel
JOUVE Hubert
NICOLAU Yvan
NIFENECKER Hervé
PERROUD Paul
PEUZIN Jean-Claude
TAIB Maurice
VINCENDON Marc

Laboratoires extérieurs

C.N.E.T

DEVINE Rodericq
GERBER Roland
MERCKEL Gérard
PAULEAU Yves

Abstract

An *equational formula* is a first order formula whose only predicate symbol is equality. We propose some transformation rules for such formulas and study their correctness in various models. Then, we investigate some controls on these rules that allow to derive termination and completeness results with respect to *solved forms*. We consider several definitions of a solved form.

As a consequence, it is possible, for example, to decide the validity of an equational formula in the Herbrand Universe. In other words, we propose a complete axiomatization of finite trees over a finite alphabet. Moreover, the termination results are given for a "minimal" control. Therefore, many *disunification* algorithms can be derived by strenghtening the control.

The above results are extended to rational trees, to order-sorted algebras and to some equational theories.

We investigate some applications. The main one is the study of correctness in algebraic specifications. More precisely, we show how it is possible to compute a (conditional) grammar for the language of irreducible ground instances of a term t (w.r.t. a given term rewriting system). Then, we propose an algorithm for "cleaning" such grammars. In particular, this shows that emptyness (and thus so-called *inductive reducibility*) is decidable in this way.

Résumé

Une *formule équationnelle* est une formule du premier ordre dont le seul symbole de prédicat est l'égalité. Nous donnons un ensemble de règles de transformation de telles formules et étudions leur correction dans divers modèles. Nous étudions ensuite plusieurs contrôles sur ces règles qui permettent d'établir des résultats de terminaison et de complétude vis à vis d'ensembles de *formes résolues*. Plusieurs notions de formes résolues sont envisagées.

Une conséquence de ces résultats est, par exemple, la décidabilité de la validité dans l'univers de Herbrand d'une formule équationnelle quelconque. En d'autres termes, nous proposons une axiomatisation complète des arbres finis sur un alphabet fini. De plus les résultats de terminaison sont établis pour un contrôle "minimal" et plusieurs algorithmes de *disunification* peuvent être obtenus par raffinement de ce contrôle.

Les résultats précédents sont étendus aux arbres rationnels, aux algèbres avec sortes ordonnées et à certaines théories équationnelles.

Nous nous intéressons à plusieurs applications. La principale d'entre elles étant l'étude du problème de la correction des spécifications algébriques et plus précisément la décision de la réductibilité inductive. Nous montrons comment, à l'aide la simplification de certaines formules équationnelles, il est possible de calculer une grammaire (conditionnelle) du langage des termes fermés irréductibles pour un système de réécriture. Nous proposons ensuite un algorithme de décision du vide pour de telles grammaires, obtenant ainsi un algorithme de décision de la réductibilité inductive dans le cas général.

Remerciements

Je remercie sincèrement tous les membres du jury:

J. Calmet qui a dirigé cette thèse et sans qui ce travail n'aurait pas vu le jour. Ses encouragements au début de ce travail m'ont été précieux pour l'entreprendre.

A. Colmerauer qui a bien voulu s'intéresser à cette thèse et dont les travaux furent des documents de travail précieux.

J. Gallier qui a bien voulu accepter la rude tache de rapporteur.

G. Huet qui a bien voulu s'intéresser à ce travail et participer au jury.

JP. Jouannaud qui m'a constamment encouragé et guidé au cours de cette thèse. Ses critiques et suggestions ont constitué un apport essentiel à mon travail.

C. Kirchner qui s'est intéressé à ce sujet et m'a toujours écouté et encouragé.

P. Lescanne avec lequel j'ai eu de très nombreuses discussions. Ses remarques et critiques m'ont été extrèmement utiles tout au long de la thèse.

JP. Verjus qui a accepté de présider ce jury.

Je remercie aussi tous les membres du LIFIA et en particulier Ph. Jorrand qui m'a accueilli dans son laboratoire et a toujours encouragé mes initiatives. Je remercie aussi mes compagnons de bureau D. Lugiez et Ph. Schnoebelen qui ont souvent eu à subir mes élucubrations. Je remercie G. Veillon, directeur de l'établissement dans lequel j'ai enseigné durant la préparation de cette thèse. Je remercie les chercheurs du CRIN et en particulier JL. Rémy pour l'intérêt qu'ils ont porté à mon travail.

Je remercie enfin et surtout mon épouse Nathalie pour le soutien qu'elle m'a apporté durant ces trois années de travail continu.

Table des matières

Chapitre 1

Introduction

La résolution de systèmes d'équations (appelée couramment unification) est un outil de plus en plus utilisé en informatique. Pour ne donner que quelques exemples d'utilisation, citons la programmation logique [Llo84], la programmation fonctionnelle [Pey87], la sémantique des langages de programmation [Sto77], l'inférence de types [Mil78] et, de façon générale, la démonstration automatique [Sti86]...

L'idée de base du travail présenté ici est d'étudier les systèmes d'équations et de "diséquations", comme, par exemple, $\{f(a,x) = f(y,y), y \neq x\}$ d'une façon symétrique. L'idée d'une telle étude n'est pas nouvelle puisque les diséquations ont été introduites en PROLOG II par A. Colmerauer [Col84]. On peut en fait s'étonner qu'elles n'aient pas été étudiées plus tôt.

Les raisons qui m'ont poussé à entreprendre cette étude sont assez différentes de celles d' A. Colmerauer. Je m'intéressais en effet aux problèmes de correction de spécifications algébriques et, plus particulièrement, au problème de savoir si une fonction f est "complètement définie" par un ensemble d'équations. Un tel ensemble d'équations engendre une congruence $=_E$ dans l'algèbre des termes. Une fonction f sera "complètement définie" par rapport un ensemble de fonctions C si tout terme contenant une occurrence de f est congru (modulo $=_E$) à un terme construit seulement avec les fonction de C. Afin de décider de la complète définition, on commence par orienter les équations de E de façon à obtenir un système de réécriture. Puis on cherche les termes de la forme $f(t_1, \ldots, t_n)$ qui ne sont pas filtrés par le membre gauche d'une règle. Cela peut être vu comme la recherche d'un contre-exemple à la complète définition de f. La fonction f est alors complètement définie s'il n'y a aucune solution à ce problème. Cette idée de base avait déjà été émise par JJ. Thiel [Thi84,LLT86].

La recherche des termes qui ne sont pas filtrés par un ensemble de termes fixés est un problème qui s'exprime bien à l'aide d'équations et de diséquations. Ceci m'a amené à étudier les systèmes d'équations et de diséquations, mais, à la différence de A. Colmerauer, ces systèmes contiennent des *paramètres* (variables quantifiées universellement). Par exemple, soit T l'ensemble des termes sans variable construits à l'aide des deux symboles de fonction : 0 (constante) et s (successeur). Une interprétation de cet ensemble de termes est l'ensemble des entiers naturels N. L'expression $P : \forall y : x \neq s(y)$ est un problème

équationnel d'inconnue x et de paramètre y. Ses solutions (dans T) sont les "valeurs" de x qui ne sont successeurs d'aucun élément de T. (Ici il n'y a bien sûr que la solution $x = 0$). Si l'on préfère, *résoudre* P c'est trouver l'ensemble $\{t \in T | \forall u \in T, t \neq s(u)\}$.

Il restait alors à donner un cadre formel à ces notions, cadre qui recouvre à la fois les problèmes connus et les nouveaux concepts introduits. C'est ce que nous faisons dans le chapitre 2 de cet ouvrage en définissant l'algèbre des *problèmes équationnels* ainsi que la notion de "solution". Nous nous inspirons pour cela des définitions données dans [KL87]. Un problème équationnel est, en bref, une formule $\exists w_1, \ldots, w_n, \forall y_1, \ldots y_m : P$ où P est une formule sans quantificateur dont les atomes sont des équations ou des diséquations. Cette notion généralise les systèmes d'équations et de diséquations de A. Colmerauer de plusieurs façons, en particulier par l'introduction des quantificateurs.

Après avoir défini les problèmes équationnels et leur sémantique, nous nous intéressons à leur "simplification". Nous donnons ainsi (chapitre 3) des règles de transformation préservant l'ensemble des solutions. Par exemple, la fusion de diséquations $t \neq u \vee t \neq v \mapsto t \neq u \vee u \neq v$. L'intention de telles règles de transformation est de simplifier les formules. Mais on voit bien sur cet exemple que la simplification n'est pas évidente. Les règles de transformation sont ainsi données dans un cadre général et leur correction est étudiée indépendamment du caractère simplificateur. Ensuite, nous proposons un *contrôle* qui, restreignant[1] les cas d'application, permet de donner des résultats de terminaison. Ce contrôle est le moins restrictif possible[2]. De cette façon on peut proposer différents algorithmes, par spécialisation du contrôle, qui tous terminent sans qu'il soit nécessaire de faire une nouvelle preuve. On peut retrouver ainsi les algorithmes d'unification de Herbrand [Her30], Martelli-Montanari [MM82], l'algorithme de Colmerauer [Col84]. On obtient aussi l'algorithme donné dans [LM87] par raffinement du contrôle. (Voir chapitre 4).

Un autre aspect très important est la nature des formes irréductibles obtenues par ces transformations. Nous envisagerons ainsi un certain nombre de *formes résolues* qui sont des problèmes plus simples. Par exemple, dans le cas de l'unification, un système d'équations dont les membres gauches sont des variables et les membres droits des termes dans lesquels n'apparaissent pas les membres gauche peut être considéré comme une forme résolue. Un tel système d'équations définit en effet un plus général unificateur. Exemple : $x_1 = f(z_1) \wedge x_2 = f(z_2)$ est en forme résolue et définit le plus général unificateur $\{x_1 \rightarrow f(z_1); x_2 \rightarrow f(z_2)\}$. Dans [Col84], les formes résolues possèdent la propriété essentielle d'avoir au moins une solution (mais on n'obtient pas toujours explicitement cette solution). Par exemple : $x_1 = f(z_1) \wedge x_2 = f(z_2) \wedge z_1 \neq z_2$.

Les formes résolues peuvent ainsi être plus ou moins résolues. On peut en effet s'intéresser à l'existence d'une solution (nous avons vu que c'était le problème qui nous intéressait dans la complétude des définitions) ou à obtenir toutes les solutions, ou un "ensemble complet" de solutions. Par exemple, en programmation logique, on repoussera le test d'occurrence autant que possible, car c'est une opération coûteuse. Les formes

[1] Par exemple, à l'aide d'un ordonnancement

[2] c'est-à-dire, par exemple, imposant un ordonnancement le plus grossier possible.

résolues sont alors différentes (cf [Hue76,Col84]). Selon le problème que l'on attaque, il peut être ainsi inutile de "pousser trop loin" la simplification. Si bien que nous structurerons les simplifications en plusieurs étapes, donnant différents contrôles et différents résultats de complétude, selon la nature des formes résolues considérées et tenant compte de la spécificité éventuelle des problèmes posés.

On peut déduire de ces résultats un algorithme de décision pour la validité dans l'univers de Herbrand d'une formule du premier ordre dont le seul symbole de prédicat est l'égalité. Ce résultat de décidabilité est nouveau. Il a été découvert indépendamment à l'aide d'une autre méthode par MJ. Maher [Mah88a].

Revenant à la motivation initiale de l'introduction des problèmes équationnels, il s'avère que le problème de la complétude des définitions n'est qu'un cas particulier du problème de la *réductibilité inductive*. Décider si un terme est inductivement réductible constitue une étape cruciale dans l'automatisation des preuves par induction dans les théorie équationnelles [JK86b]. Il est donc naturel de se poser le problème de l'extension possible à la réductibilité inductive de la méthode employée pour la complète définition. Ce travail m'a amené alors (sur une idée de JL Rémy) à introduire le *langage des formes normales fermées* et à résoudre certains problèmes équationnels permettant de calculer une grammaire de ce langage. Ces grammaires sont extrêmement utiles car elles fournissent un outil de construction systématique de représentants dans les algèbres quotient. Comme le problème de la réductibilité inductive est équivalent au problème de la décision du vide d'un certain langage de formes normales fermées, nous nous intéressons dans le chapitre 5 au calcul et aux propriétés de ces grammaires. Nous donnons en particulier un algorithme de décision du vide pour les langages de formes normales.

La simplification des problèmes équationnels (telle qu'elle est développée dans les chapitres 3 et 4) possède de très nombreuses applications, le problème de la réductibilité inductive n'en étant qu'un exemple. Parmi les applications déjà développées, nous pouvons citer l'analyse temporelle des processus communicants en FP2 [Jor86] et la compilation du filtrage [Sch87b,Sch88b]. Les problèmes équationnels sont aussi utilisés pour le traitement de la négation explicite en programmation logique. Plus généralement les problèmes équationnels sont utilisés dans la transformation des programmes logiques [Lug88] et dans l'expression du contrôle [JL87].

D'autre part, le problème du choix de la forme résolue pour exprimer un ensemble de solutions devient crucial dans des problèmes d'implantation effective, par exemple pour l'unification Associative Commutative [Sti81]. Le nombre d'unificateurs minimaux peut en effet se révéler très (trop) élevé même sur des exemples très simples [Bur88]. HJ. Bürckert a alors eu l'idée de représenter ces solutions dans une théorie plus large (AC1) en contraignant les variables. Cela se traduit par la résolution d'un problème équationnel avec des formes résolues particulières [Bur88]. Cette direction de recherche semble prometteuse pour la théorie de l'unification comme pour les problèmes de complétion des systèmes de réécriture [KK88].

Les grammaires de formes normales sont, elles-aussi, utilisés dans d'autres applications. Par exemple pour transformer une spécification contenant des "relations entre constructeurs" en une spécification "sans relations entre constructeurs", transformation préservant

les théorèmes inductifs [Com88a].

Je ne présente néanmoins dans cette thèse que les applications que j'ai personnellement étudiées. Il s'agit tout d'abord de l'utilisation des problèmes équationnels pour traiter de façon explicite la négation en programmation logique. La simplification des problèmes équationnels permet en effet de calculer la "contrepartie négative" d'un programme et permet de traiter de façon symétrique un prédicat et sa négation. Cette même idée est d'ailleurs présentée dans l'article de Barbuti & all. [BMPT87]. Enfin, cherchant une définition complète de l'égalité sur les entiers relatifs pour mes problèmes de complétude des définitions, je me suis aperçu qu'il était impossible d'en donner une sans introduire de "fonction cachée". Le problème peut néanmoins être contourné dans ce cas (et bien d'autres) en utilisant une grammaire du langage des formes normales fermées. Il suffit en effet d'associer une sorte à chaque non terminal de la grammaire pour obtenir une présentation avec sortes ordonnées, dans laquelle il n'y a plus de relation entre les fonctions de la signature. Ce résultat renforce celui de Goguen et Meseguer [GM87b] sur la puissance d'expression des algèbres avec sortes ordonnées, par rapport aux algèbres multi-sortes.

Toutes ces applications sont intervenues "après coup". Elles sont présentées dans le chapitre 6 de cette thèse. Leur liste n'est certainement pas close.

Les résultats de terminaison et de complétude présentés dans les chapitres3 et 4 font essentiellement référence à des théories "libres" c'est-à-dire supposent l'absence d'axiomes reliant les symboles fonctionnels. Par ailleurs de nombreux travaux sur l'unification (par exemple [Kir85]) ont précisément pour objet l'étude de l'unification dans les théories équationnelles, c'est-à-dire en présence d'axiomes donnés sous forme d'équations. L'objet du chapitre 7 est donc d'étendre autant que possible les résultats précédents aux théories équationnelles. Nous montrons en fait que ces résultats s'étendent aux théories "quasi-libres", mais le problème reste ouvert dans le cas où les axiomes sont constitués de la commutativité et de l'associativité de certaines fonctions.

Enfin, le cas des théories équationnelles se posait de même pour l'extension des résultats du chapitre 5. Dans le cas de théories équationnelle, la notion de réductibilité inductive est modifiée. En particulier, si l'on utilise la complétion sans échec [HR87] (qui semble bien adaptée à l'automatisation des preuves par induction), la réductibilité inductive s'exprime bien à l'aide d'équations, de diséquations et d'*inéquations*. L'étude des systèmes comportant aussi des inéquations est une question ouverte dont nous présenterons une ébauche dans le chapitre 8.

Exemples introductifs

Cette section a pour but d'introduire à notre problématique à l'aide de trois exemples simples. Le premier est un problème d'unification très simple, le second est un problème de résolution d'équations et de diséquations, tel qu'il se poserait en PROLOG II, le troisième est un exemple de problème issu des types abstraits algébriques. Ces trois exemples constituent des cas très particuliers des problèmes étudiés dans les chapitres suivants.

Nous ne donnerons pas ici de définitions générales (c'est l'objet du chapitre suivant), mais seulement le minimum nécessaire à la compréhension des exemples. De plus, aucune référence à ce chapitre ne sera faite dans la suite. Le lecteur averti peut donc se passer de sa lecture. Aucune preuve n'est donnée ici.

Dans la suite, F désignera un ensemble de symboles fonctionnels, X un ensemble de variables, $T(F)$ et $T(F,X)$ les ensembles de termes construits respectivement sur F et sur F et X. Une *équation* est une paire de termes (s,t) notée $s = t$. Une *diséquation* est une paire de termes (s,t) notée $s \neq t$. Les équations comme les diséquations ne sont pas orientées. Si bien que $s = t$ et $t = s$ désignent la même équation. $\top$ désigne la classe des équations $s = s$ et $\bot$ la classe des diséquations $s \neq s$.

Afin d'éviter les confusions, les symboles $=$ et $\neq$ seront réservés aux équations et diséquations. Nous noterons ainsi $\equiv$ et $\not\equiv$ l'égalité et l'inégalité syntaxique respectivement.

$\wedge$ et $\vee$ désigneront respectivement la conjonction et la disjonction. Ces opérations sont supposées satisfaire les propriétés bien connues des algèbres booléennes.

Une *substitution* σ associe à un ensemble fini de variables appelé *domaine* de la substitution et noté $Dom(\sigma)$ un ensemble fini de termes de $T(F,X)$. Une substitution peut être prolongée de façon unique en un endomorphisme de $T(F,X)$ tel que $x\sigma \equiv x$ pour $x \notin Dom(\sigma)$. Une substitution est dite *fermée* si les images des variables de son domaine sont dans $T(F)$.

Un *problème équationnel* est - ou bien une équation -ou bien une diséquation -ou bien une expression $\mathcal{P}_1 \vee \mathcal{P}_2$ ou $\mathcal{P}_1 \wedge \mathcal{P}_2$ où $\mathcal{P}_1$ et $\mathcal{P}_2$ sont des problèmes équationnels.

Bien sûr, il y a plusieurs représentants pour un même problème équationnel (par exemple, la forme normale disjonctive ou la forme normale conjonctive). Nous nous permettrons de choisir le représentant qui nous convient, en fonction du contexte.

Une substitution fermée σ *valide* un problème $\mathcal{P}$ si l'une des propriétés suivantes est satisfaite:

- $\mathcal{P} \equiv \top$
- $\mathcal{P}$ est une équation $s = t$ et $s\sigma \equiv t\sigma$
- $\mathcal{P}$ est une diséquation $s \neq t$ et $s\sigma \not\equiv t\sigma$
- $\mathcal{P}$ est de la forme $\mathcal{P}_1 \vee \mathcal{P}_2$ et σ valide $\mathcal{P}_1$ ou $\mathcal{P}_2$
- $\mathcal{P}$ est de la forme $\mathcal{P}_1 \wedge \mathcal{P}_2$ et σ valide $\mathcal{P}_1$ et $\mathcal{P}_2$

Une substitution σ est une *solution* d'un problème équationnel $\mathcal{P}$ si

- σ est une substitution fermée dont le domaine est l'ensemble des variables de $\mathcal{P}$
- σ valide $\mathcal{P}$.

Notre objectif est de transformer les problèmes équationnels en des formes plus simples (formes résolues) qui ont les mêmes solutions. Montrons sur trois exemples ce que peuvent être des problèmes équationnels, ce que peuvent être des règles de transformation et ce que peuvent être des formes résolues.

Exemple 1.1 Le problème équationnel que nous considérons ici est réduit à la seule équation:

$$f(x, g(y)) = f(g(z), x)$$

Nous dirons que c'est un problème *d'inconnues* x, y *et* z. La première règle de transformation que nous utilisons est la *décomposition.* Cette règle consiste simplement à simplifier par le symbole de tête et peut s'énoncer:

$$(D_1) \quad f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n) \quad \mapsto \quad t_1 = u_1 \wedge \ldots t_n = u_n$$

On en déduit ainsi que:

$$f(x, g(y)) = f(g(z), x) \quad \mapsto_{D_1} \quad x = g(z) \wedge g(y) = x$$

Nous pouvons ensuite tirer parti de la transitivité de la relation $=$ en effectuant une *fusion*, qui correspond à la règle:

$$(F_1) \quad x = u \wedge x = v \quad \mapsto \quad x = u \wedge u = v$$

Ainsi, le problème peut à nouveau être transformé :

$$x = g(z) \wedge g(y) = x \quad \mapsto_{F_1} \quad x = g(z) \wedge g(z) = g(y) \quad \mapsto_{D_1} \quad x = g(z) \wedge y = z$$

Ce dernier problème est en forme résolue. Il nous donne en effet une "plus générale substitution" $\sigma = \{x \to g(z);\ y \to z\}$ qui permet d'unifier les deux termes de l'équation de départ. L'ensemble de toutes les solutions s'obtient alors en composant σ avec une substitution quelconque sur z.

Exemple 1.2 Dans ce nouvel exemple, nous introduisons d'une part des diséquations et d'autre part des disjonctions. La méthode de transformation employée est alors essentiellement la même que celle de A. Colmerauer [Col82,Col84].

$$\mathcal{P} \quad \equiv \quad (x \neq b \vee f(x, y) \neq f(x, x)) \wedge y = g(x)$$

Ici, x et y sont les inconnues du problème et b est une constante.

De même que nous avons décomposé les équations dans l'exemple précédent, nous pouvons décomposer les diséquations en utilisant la règle:

$$(D_2) \quad f(t_1,\ldots,t_n) \neq f(u_1,\ldots,u_n) \quad \mapsto \quad t_1 \neq u_1 \vee \ldots t_n \neq u_n$$

On en déduit ainsi la transformation:

$$\mathcal{P} \quad \mapsto_{D_2} \quad (x \neq b \vee x \neq x \vee y \neq x) \wedge y = g(x)$$

La diséquation $x \neq x$ n'est jamais satisfaite et on peut ainsi l'éliminer, utilisant la règle:

$$(T_2) \quad s \neq s \quad \mapsto \quad \bot$$

$$(x \neq b \vee x \neq x \vee y \neq x) \wedge y = g(x) \quad \mapsto_{T_2} \quad (x \neq b \vee y \neq x) \wedge y = g(x)$$

Remarquons que nous avons aussi utilisé le fait que $\bot$ est élément neutre de $\vee$.

On peut maintenant fusionner les diséquations comme nous avions fusionné les équations, utilisant la règle

$$(F_2) \quad x \neq u \vee x \neq v \quad \mapsto \quad x \neq u \vee u \neq v$$

Il est aussi possible de fusionner équations et diséquations, utilisant la règle:

$$(F_3) \quad x = t \wedge (x \neq u \vee d) \quad \mapsto \quad x = t \wedge (t \neq u \vee d)$$

On obtient ainsi les transformations:

$$\begin{aligned} (x \neq b \vee y \neq x) \wedge y = g(x) \quad &\mapsto_{F_2} \quad (x \neq b \vee y \neq b) \wedge y = g(x) \\ &\mapsto_{F_3} \quad (x \neq b \vee g(x) \neq b) \wedge y = g(x) \end{aligned}$$

Mais la diséquation $g(x) \neq b$ est toujours satisfaite (dans l'algèbre des termes), ce que l'on modélise par la règle d'incompatibilité suivante:

$$(I_2) \quad f(t_1,\ldots,t_n) \neq g(u_1,\ldots,u_m) \quad \mapsto \quad \top \quad \textit{Si } f \textit{ et } g \textit{ sont distincts}$$

Utilisant alors cette règle et la propriété d'élément absorbant de $\top$ pour $\vee$, on obtient:

$$(x \neq b \vee g(x) \neq b) \wedge y = g(x) \quad \mapsto_{I_2} \quad y = g(x)$$

Ce dernier problème est en "forme résolue"[3]. Comme chaque transformation conserve l'ensemble des solutions, nous pouvons dire que les solutions de $\mathcal{P}$ sont de la forme $\{x \rightarrow t;\ y \rightarrow g(t)\}$ où t est un terme de $T(F)$ quelconque.

[3]Une forme résolue est ici un système d'équations définissant un plus général unificateur.

Dans cet exemple nous avons obtenu à nouveau un plus général unificateur comme forme résolue. Mais ce n'est pas toujours le cas lorsque l'on part d'un système d'équations et de diséquations[4].

Exemple 1.3 Dans cet exemple, nous allons généraliser un peu la notion de problème équationnel en autorisant la quantification universelle de certaines variables. Cette extension permettra d'exprimer le problème de complétude de définitions comme dans l'exemple suivant:

$F = \{0 : \rightarrow nat;\ \ s : nat \rightarrow nat;\ \ + : nat \times nat \rightarrow nat\}$

$+$ est défini par le système de réécriture:[5]

$$\begin{array}{rcl} y_1 + 0 & \rightarrow & y_1 \\ y_1 + s(y_2) & \rightarrow & s(y_1 + y_2) \end{array}$$

$+$ est complètement défini si et seulement si tout terme de $T(F)$ a une forme irréductible dans $T(\{0, s\})$. Inversement, $+$ n'est pas complètement défini si on peut trouver des instances de $x_1 + x_2$ qui ne sont pas filtrées par un des membres gauches de règle. Ce que l'on peut exprimer à l'aide de la formule:

$$\exists x_1, x_2 \in T(\{0, s\}),\ \forall y_1, y_2 \in T(\{0, s\}),\ \ x_1 + x_2 \neq y_1 + 0 \wedge x_1 + x_2 \neq y_1 + s(y_2)$$

Nous autoriserons donc désormais la quantification universelle de certaines variables d'un problème équationnel. Ces variables seront appelées *paramètres* alors que les variables non quantifiées (variables libres) seront les *inconnues* du problème.

Une solution d'une telle formule dans $T(C)$ où C est un sous-ensemble de F ($C = \{0, s\}$ dans l'exemple) est une substitution σ dont le domaine est l'ensemble des inconnues et telle que -l'image des inconnues est dans $T(C)$ - pour toute substitution θ sur les paramètres du problème, $\sigma\theta$ valide P.

En termes de problèmes équationnels, $+$ n'est pas complètement défini ssi le problème:

$$\mathcal{P} \equiv \forall y_1, y_2 : x_1 + x_2 \neq y_1 + 0 \wedge x_1 + x_2 \neq y_1 + s(y_2)$$

a au moins une solution dans $T(\{0, s\})$.

[4] Le problème de la transformation des diséquations en équations et en particulier la caractérisation des cas où cela est possible sera abordé dans le chapitre 4.

[5] $+$ étant binaire, nous l'utilisons en notation infixée

Tout comme dans les exemples précédents, on peut commencer par décomposer les diséquations[6]:

$$\forall y_1, y_2 : x_1 + x_2 \neq y_1 + 0 \wedge x_1 + x_2 \neq y_1 + s(y_2)$$
$$\mapsto_{D_2} \forall y_1, y_2 : (x_1 \neq y_1 \vee x_2 \neq 0) \wedge (x_1 \neq y_1 \vee x_2 \neq s(y_2))$$

Nous avons besoin maintenant de nouvelles règles de transformation qui permettent d'*éliminer les paramètres.* Remarquons que, quelle que soit la substitution sur x_1, la condition $x_1 \neq y_1$ ne peut être satisfaite puisque la valeur correspondante de y_1 invalide la diséquation. Ceci s'exprime à l'aide de la règle :

$$(U_2) \quad y \neq t \vee d \quad \mapsto \quad d\{y \to t\} \quad \textit{Si } y \textit{ est un paramètre}$$

d désigne ici n'importe quelle formule et $d\{y \to t\}$ désigne la formule d dans laquelle on a remplacé toutes les occurrences de y par t.

Alors,

$$\forall y_1, y_2 : (x_1 \neq y_1 \vee x_2 \neq 0) \wedge (x_1 \neq y_1 \vee x_2 \neq s(y_2)) \mapsto_{U_2} \forall y_1, y_2 : x_2 \neq 0 \wedge x_2 \neq s(y_2)$$

Maintenant, y_1 n'apparaissant plus dans le corps du problème, peut être éliminée. Pour éliminer le deuxième paramètre il faut par contre utiliser le fait que x_2 ne prend ses valeurs que dans $T(C)$ en faisant une "décomposition par cas". Une telle règle peut s'écrire:

$$(Ex) \quad P \wedge x \neq u \quad \mapsto \quad \bigvee_{f \in C} (\exists z_1, \ldots, z_m : \ (P \wedge x \neq u) \wedge x = f(z_1, \ldots z_m))$$

où $z_1, \ldots, z_m$ sont des variables qui n'apparaissaient pas dans le problème. Bien sûr, cette règle ne doit pas être utilisée systématiquement, mais seulement lorsque u contient une occurrence de paramètre et qu'aucune autre règle n'est applicable.

On peut maintenant terminer la résolution du problème posé:

$$\forall y_2 : x_2 \neq 0 \wedge x_2 \neq s(y_2)$$
$$\mapsto_{Ex} (\forall y_2 : x_2 \neq 0 \wedge x_2 \neq s(y_2) \wedge x_2 = 0) \vee (\exists z_1, \forall y_2 : x_2 \neq 0 \wedge x_2 \neq s(y_2) \wedge x_2 = s(z_1))$$
$$\mapsto_{F_3} (\forall y_2 : 0 \neq 0 \wedge 0 \neq s(y_2) \wedge x_2 = 0) \vee (\exists z_1, \forall y_2 : s(z_1) \neq 0 \wedge s(z_1) \neq s(y_2) \wedge x_2 = s(z_1))$$
$$\mapsto_{T_2} \exists z_1, \forall y_2 : s(z_1) \neq 0 \wedge s(z_1) \neq s(y_2) \wedge x_2 = s(z_1))$$
$$\mapsto_{T_2} \exists z_1, \forall y_2 : s(z_1) \neq s(y_2) \wedge x_2 = s(z_1))$$
$$\mapsto_{D_2} \exists z_1, \forall y_2 : z_1 \neq y_2 \wedge x_2 = s(z_1))$$
$$\mapsto_{U_2} \bot$$

[6] Bien sûr, cette transformation est indépendante des propriétés de $+$.

Ce qui nous prouve qu'il n'y a pas de solution au problème initial. Et donc que + est complètement défini.

Notons que nous avons quand même un peu triché en introduisant des quantificateurs existentiels sans avoir défini ce qu'est une solution dans ce cas. Tout cela sera précisé dans la section suivante et nous avons voulu éviter trop d'interruptions par des définitions que chacun peut deviner.

Chapitre 2

Problèmes équationnels

2.1 Définitions préliminaires et notations

Les notations que nous employons sont, la plupart du temps, celles de G. Huet et D. Oppen [HO80] et (sauf mention contraire) celles données par Dershowitz et Jouannaud [DJ88].

2.1.1 F-algèbres

Une *signature* est un couple (S, F) où S est un ensemble fini dont les éléments sont appelés *sortes* et F est un ensemble de *symboles fonctionnels* muni d'une fonction de typage τ qui associe à chaque élément de F une séquence non vide d'éléments de S. On écrira $f : \underline{s}_1 \times \ldots \times \underline{s}_n \rightarrow \underline{s}$ à la place de $\tau(f) = (\underline{s}_1, \ldots, \underline{s}_n, \underline{s})$. n est appelé *arité* de f. Les symboles d'arité 0 sont appelés *constantes*. Sauf précision contraire, F sera supposé fini.

Etant donné une signature (S, F), une *F-algèbre* $\mathcal{A}$ est une famille d'ensembles $\{D_{\mathcal{A},s} | s \in S\}$ munie d'une famille d'applications $\{f_{\mathcal{A}} | f \in F\}$ telles que, si $f : \underline{s}_1 \times \ldots \times \underline{s}_n \rightarrow \underline{s}$, $f_{\mathcal{A}}$ est une application de $D_{\mathcal{A},\underline{s}_1} \times \ldots \times D_{\mathcal{A},\underline{s}_n}$ dans $D_{\mathcal{A},\underline{s}}$. Pour toute sorte $\underline{s} \in S$, l'ensemble $D_{\mathcal{A},\underline{s}}$ est appelé *support* de $\underline{s}$ dans $\mathcal{A}$. Si $A \subseteq \mathcal{A}$, le *support dans* A de $\underline{s}$ est l'ensemble $A \cap D_{\mathcal{A},\underline{s}}$.

Pour tout élément $\underline{s} \in S$, $X_{\underline{s}}$ est un ensemble infini dénombrable de symboles disjoints de ceux de F et disjoints entre eux. X est la réunion des $X_{\underline{s}}$. Ces symboles sont appelés *variables*.

$T(F, X)$ désigne alors l'ensemble des termes "bien formés" sur la signature (S, F) (S est souvent omis) et l'ensemble de variables X. C'est la F-algèbre hétérogène libre de générateurs X.
$T(F)$ désigne la F-algèbre libre sur 0 générateur. Nous supposerons qu'il existe dans $T(F)$ au moins un terme de chaque sorte. $T(F)$ est aussi une algèbre initiale dans la catégorie des F-algèbres. (Les autres lui sont isomorphes).

Un élément de $T(F, X)$ ou de $T(F)$ est appelé *terme*. Un terme $t \in T(F, X)$ est dit *linéaire* si toute variable apparait au plus une fois dans t. Sa sorte est notée $sort(t)$. Lorsque $sort(t)$ a un support infini dans $T(F)$, on dira que t (resp. $sort(t)$) est *infinitaire*.

Dans le cas contraire t est dit *finitaire.* ,

Un ensemble de positions **Pos** est un ensemble de mots d'entiers naturels supérieurs ou égaux à 1 (la concaténation est notée $\cdot$ et le mot vide ϵ) qui vérifie:

- $\forall u \cdot v \in \mathbf{Pos}$, $u \in \mathbf{Pos}$ (stabilité par préfixe)
- $\forall u \cdot i \in \mathbf{Pos}, \forall j, 1 \leq j \leq i$, $u \cdot j \in \mathbf{Pos}$

Un arbre étiqueté est alors une application t associant à un ensemble de positions un ensemble d'étiquettes. Un terme t de $T(F, X)$ est un arbre étiqueté dont l'ensemble des positions est noté $Pos(t)$, l'ensemble des étiquettes est contenu dans $F \cup X$ et défini comme suit:

- Si t est une variable ou une constante, alors $Pos(t) = \{\epsilon\}$ et $t(\epsilon) = t$
- Si $t = f(t_1, \ldots, t_n)$, alors $Pos(t) = \bigcup_{1 \leq i \leq n} \{i \cdot u | u \in Pos(t_i)\}$, $t(\epsilon) = f$ et $t(i \cdot u) = t_i(u)$.

Si t est un terme et $p \in Pos(t)$, le *sous-terme* de t à la position p est noté t/p et est défini par :

- $Pos(t/p) = \{q | p \cdot q \in Pos(t)\}$
- Pour tout $q \in Pos(t/p)$, $t/p(q) = t(p \cdot q)$

Si t, u sont deux termes, on note $t[u]_p$ le terme obtenu en remplaçant t/p par u dans t:

- $Pos(t[u]_p) = \{q \in Pos(t) | \forall q', q \neq p \cdot q'\} \cup \{p \cdot q | q \in Pos(u)\}$
- Si $q \in Pos(t[u]_p)$, - ou bien $q = p \cdot q'$ et $t[u]_p(q) = u(q')$ - ou bien q n'est pas suffixe de p et $t[u]_p(q) = t(q)$.

Si u est déjà un sous-terme de t on notera simplement $t[u]$.

Deux positions p et p' sont *disjointes* si elles sont incomparables pour l'ordre lexicographique. Lorsque $p_1, \ldots, p_n$ sont des positions disjointes deux à deux d'un terme t, on définit par récurrence le *remplacement multiple* dans t aux positions $p_1, \ldots, p_n$ par les termes $u_1, \ldots, u_n$:

$$t[u_1, \ldots, u_n]_{p_1, \ldots, p_n} = (t[u_1, \ldots, u_{n-1}]_{p_1, \ldots, p_{n-1}})[u_n]_{p_n}$$

La *profondeur* d'un terme t le nombre $\max\{|p|,\ p \in Pos(t)\}$ si $|p|$ est la longueur de la position p (considérée comme une séquence). La *taille* d'un terme t (parfois notée $|t|$) est le cardinal de $Pos(t)$.

2.1.2 Substitutions

Σ désigne l'ensemble des morphismes de F-algèbres de $T(F,X)$ dans lui-même. Un élément de Σ est appelé *substitution* [HO80]. Si $\sigma \in \Sigma$, le *domaine* de σ est défini par[1]:

$$Dom(\sigma) = \{x \in X, x\sigma \neq x\}$$

En fait, nous ne nous intéresserons qu'au cas où $Dom(\sigma)$ est fini. Par abus de langage, nous désignerons encore par Σ l'ensemble des substitutions dont le domaine est fini et nous omettrons l'attribut "de domaine fini" en parlant des substitutions.

Plus généralement, si $\mathcal{A}$ est une F-algèbre, nous appellerons *$\mathcal{A}$-substitution* tout homomorphisme de F-algèbre σ dont l'ensemble de départ est $T(F,X_0)$ et l'ensemble d'arrivée est $\mathcal{A}$, X_0 étant un sous-ensemble fini de X encore noté $Dom(\sigma)$.[2]

Lorsque $\mathcal{A}$ est un sous-ensemble de $T(F,X)$, une $\mathcal{A}$-substitution σ s'étend en un morphisme de $T(F,X)$ dans lui-même en ajoutant les conditions : $x\sigma = x$, pour tout $x \in X - Dom(\sigma)$. Les notions de $T(F,X)$-substitution et de substitution coincident alors. C'est pourquoi nous emploierons indifféremment les deux terminologies.

Σ_g (g pour "ground") désignera l'ensemble des $T(F)$-substitutions aussi appelées *substitutions fermées.*

Nous utiliserons également les notations suivantes:

- Si $t \in T(F,X)$, $Var(t)$ est l'ensemble des variables apparaissant dans t. Plus généralement, si e est une expression quelconque, $Var(e)$ désignera l'ensemble des variables apparaissant dans cette expression.
- Si σ est une $\mathcal{A}$-substitution où $\mathcal{A}$ est un sous-ensemble de $T(F,X)$, l'ensemble des variables de l'*image* de σ est défini par:

$$VIm(\sigma) = \{y \in X | \exists x \in Dom(\sigma), y \in Var(x\sigma)\}$$

- Si $t_1, \ldots, t_n \in \mathcal{A}$, $\{x_1 \rightarrow t_1; \ldots; x_n \rightarrow t_n\}$ désignera la $\mathcal{A}$-substitution σ définie par:
  1. $Dom(\sigma) = \{x_1, \ldots, x_n\}$
  2. $\forall i \in \{1, \ldots, n\}$, $x_i\sigma = t_i$

Définition 2.1 *X_0 étant un ensemble de variables, une $\mathcal{A}$-substitution σ sera dite* en dehors *de X_0 si $Dom(\sigma) \cap X_0 = \emptyset$ et $VIm(\sigma) \cap X_0 = \emptyset$*[3].

"o" désignera la composition habituelle des applications. Mais nous aurons également besoin dans la suite de la *juxtaposition* des substitutions. Soient σ et θ deux $\mathcal{A}$-substitutions. On note $\sigma\theta$ la $\mathcal{A}$-substitution définie par :

[1] Nous emploierons dans tout cet ouvrage la notation xf pour désigner l'application de f à x, au moins lorsque f est une substitution. Cette notation tend à s'imposer [DJ88]

[2] Notons qu'il s'agit ici d'une définition inhabituelle puisqu'une $\mathcal{A}$-substitution n'est pas définie sur tout $T(F,X)$. Mais nous pouvons maintenant substituer des éléments d'une F-algèbre quelconque tout en conservant des substitutions à domaine fini

[3] Cette dernière condition n'a de sens que si $\mathcal{A} \subseteq T(F,X)$.

- $Dom(\sigma\theta) = Dom(\sigma) \cup Dom(\theta)$
- $\forall x \in Dom(\sigma)$, $x\sigma\theta = x\sigma$
- $\forall x \in Dom(\theta) - Dom(\sigma)$, $x\sigma\theta = x\theta$

Cette définition coincide avec la composition dans tous les cas qui vont nous occuper, c'est pourquoi nous utilisons une telle notation. Plus précisément, si σ et θ sont deux substitutions telles que θ soit en dehors de $VIm(\sigma)$, alors $\theta \circ \sigma = \sigma\theta$ [4].

2.1.3 Théorie équationnelle

Rappelons brièvement la définition d'une théorie équationnelle.

Une relation binaire R sur $T(F,X)$ est *stable par substitution* si

$$\forall \sigma \in \Sigma,\ (t\ R\ s \ \Rightarrow\ t\sigma\ R\ s\sigma)$$

R est une *précongruence*[5] si

$$\forall t,s,s' \in T(F,X), \forall p \in Pos(t),\ (s\ R\ s' \ \Rightarrow\ t[s]_p\ R\ t[s']_p)$$

Etant donné un ensemble fini E de paires de termes de même sorte (t_i, u_i), $=_E$ désigne la plus petite précongruence symétrique, réflexive et transitive sur $T(F,X)$ qui soit stable par substitution et telle que, pour tout i, $t_i =_E u_i$.

Une F-algèbre $\mathcal{A}$ est un *modèle* de $=_E$ si, pour toute $\mathcal{A}$-substitution σ et tout indice i, $t_i\sigma =_{\mathcal{A}} u_i\sigma$ (les équations de E sont ainsi implicitement quantifiées universellement). La *théorie équationnelle* définie par E (nous dirons aussi *engendrée par* E) est la classe des F-algèbres qui sont des modèles de $=_E$.

Le théorème de Birkhoff [Bir35] confère un rôle central à l'algèbre $T(F,X)/=_E$ puisque les théorèmes de la théorie équationnelle sont les égalités de $T(F,X)/=_E$. Nous confondrons ainsi la théorie équationnelle et l'ensemble des équations $u = v$ telles que $u =_E v$.

Enfin, nous appelerons *spécification* (multi-sorte) tout triplet (S,F,E). Les équations de E seront parfois notées $u == v$ au lieu de $u = v$ pour faire la distinction entre "axiomes" et "équations à résoudre".

2.1.4 Systèmes de réécriture

Un système de réécriture $\mathcal{R}$ est un ensemble fini de paires orientées (l,r) formées de deux termes de même sorte et tels que $Var(r) \subseteq Var(l)$. La *relation de réduction* associée à un système de réécriture $\mathcal{R}$ est la plus petite précongruence $\rightarrow_{\mathcal{R}}$ sur $T(F,X)$ qui soit stable

[4] Cela permet en particulier d'avoir la relation $x(\sigma\theta) = (x\sigma)\theta$.

[5] On dit aussi *compatible avec la structure de F-algèbre*; quand on parle de précongruence, F est sous entendu.

par substitution et telle que $\forall (l,r) \in \mathcal{R},\ l \rightarrow_{\mathcal{R}} r$.

On dit que t se réduit en u à la position p par la règle $l \rightarrow r$ s'il existe une substitution σ telle que $t/p = l\sigma$ et $u = t[r\sigma]_p$.

On note habituellement $\rightarrow^*$ la fermeture réflexive transitive de la relation $\rightarrow$, $\rightarrow^+$ sa fermeture transitive et $\leftrightarrow$ sa fermeture symétrique.

Un système de réécriture $\mathcal{R}$ est *confluent* lorsque

$$(t \rightarrow^*_{\mathcal{R}} u \text{ et } t \rightarrow^*_{\mathcal{R}} v) \Rightarrow (\exists s,\ u \rightarrow^*_{\mathcal{R}} s \text{ et } v \rightarrow^*_{\mathcal{R}} s)$$

$\mathcal{R}$ est *noethérien* s'il n'y a pas de suite infinie t_i de termes de $T(F,X)$ telle que, pour tout i, $t_i \rightarrow_{\mathcal{R}} t_{i+1}$. $\mathcal{R}$ est *convergent* s'il est confluent et noethérien.

Lorsque $\mathcal{R}$ est convergent, tout terme t admet une forme irréductible unique notée $t\downarrow$. On a alors

$$t \leftrightarrow^* s \Leftrightarrow t\downarrow = s\downarrow$$

Un système de réécriture est dit *canonique* si, pour toute règle (l,r) de $\mathcal{R}$, r est irréductible et l n'est réductible que par la règle (l,r).

2.2 Formules équationnelles

Une formule équationnelle est une formule du premier ordre dont le seul symbole de prédicat est l'égalité. Pour moi, l'étude de telles formules a été essentiellement motivée par les problèmes de complétude suffisante. Donnons en dès maintenant une idée.

On suppose que F est scindé en deux sous-ensembles disjoints C et D. C est l'ensemble des *constructeurs* . On dispose d'autre part d'un système de réécriture $\mathcal{R}$ supposé canonique et tel que tout terme de $T(C)$ soit irréductible. Un symbole fonctionnel $f \in D$ est *complètement défini* par rapport à C si tout terme de $T(C \cup \{f\})$ a pour forme irréductible un terme de $T(C)$[6]. Ce problème est lié à la complétude suffisante des types abstraits algébriques [GH78]. Il peut s'exprimer simplement en utilisant des formules du premier ordre. Remarquons tout d'abord qu'il suffit de vérifier que, pour tous termes $t_1,\ldots t_n \in T(C)$, $f(t_1,\ldots,t_n)$ est réductible. Il suffit même de vérifier que tout terme $f(t_1,\ldots t_n)$ est filtré par un membre gauche de règle, c'est-à-dire, si $g_1,\ldots,g_k$ est l'ensemble des membres gauche de $\mathcal{R}$ et $\{x_1,\ldots,x_m\} = Var(g_1,\ldots,g_k)$:

$$\forall t_1,\ldots,t_n \in T(C),\ \exists x_1,\ldots,x_m \in T(C),\ f(t_1,\ldots t_n) = g_1 \vee \ldots \vee f(t_1,\ldots,t_n) = g_k$$

En prenant la négation de cette formule, on peut aussi dire que f est bien défini par rapport à C ssi il n'y a pas de solution dans $T(C)$ au problème :

$$\forall x_1,\ldots,x_m, f(t_1,\ldots,t_n) \neq g_1 \wedge \ldots \wedge f(t_1,\ldots,t_n) \neq g_k$$

[6]Cette question sera étudiée dans un cadre plus général au chapitre 5

Nous verrons que cette deuxième formulation est plus commode.

Revenons aux formules équationnelles.

Définition 2.2 *Une* équation *est un ensemble* $\{s,t\}$ *de termes de même sorte et notée* $s = t$. *Une équation n'est pas orientée. c'est-à-dire que* $s = t$ *et* $t = s$ *représentent la même équation.*

Afin d'éviter les confusions nous noterons désormais $\equiv$ l'égalité syntaxique des termes. Ce symbole désignera aussi dans la suite l'égalité entre formules équationnelles.

La définition qui suit est un rappel de ce qu'est une formule du premier ordre dans le cadre de notre étude [Gal86].

Définition 2.3 *Une* formule équationnelle *est*

- *ou bien une équation* $s = t$
- *ou bien l'un des deux symboles* $\bot$ *ou* $\top$
- *ou bien une expression* $P \vee Q$ *où* P *et* Q *sont des formules équationnelles*
- *ou bien une expression* $P \wedge Q$ *où* P *et* Q *sont des formules équationnelles*
- *ou bien une expression* $\exists x : P$ *où* P *est une formule équationnelle et* x *une variable*
- *ou bien une expression* $\forall x : P$ *où* P *est une formule équationnelle et* x *une variable*
- *ou bien une expression* $\neg P$ *où* P *est une formule équationnelle*

Nous adopterons la notation $s \neq t$ pour $\neg(s = t)$. $s \neq t$ est alors appelée *diséquation.* $\top$ ainsi que toutes les équations $s = s$ seront appelés *équations triviales.* $\bot$ ainsi que toutes les diséquations $s \neq s$ seront appelés *diséquations triviales.*

On définit aussi comme d'habitude la notion de *variable libre* d'une formule équationnelle (l'ensemble des variables libres de ϕ est noté $VL(\phi)$)par induction sur la structure d'une formule:

- $VL(s = t) = Var(s,t)$
- $VL(\top) = VL(\bot) = \emptyset$
- $VL(\neg P) = VL(P)$
- $VL(P \wedge Q) = VL(P \vee Q) = VL(P) \cup VL(Q)$
- $VL(\exists x : P) = VL(\forall x : P) = VL(P) - \{x\}$

Rappelons enfin la définition d'un modèle d'une formule (cf [Caf86,Gal86]):

Définition 2.4 *Une* F*-algèbre* $\mathcal{A}$ *et une* $\mathcal{A}$*-substitution* σ *constituent un* modèle *de la formule équationnelle* ϕ *(ce que l'on note* $(\mathcal{A},\sigma) \models \phi$ *ou* $\sigma \in \mathcal{S}(\mathcal{A},\phi)$*) ssi* $Dom(\sigma)$ *contient* $VL(\phi)$ *et l'une des conditions suivantes est remplie:*

- ϕ *est une équation* $s = t$ *et* $s\sigma =_{\mathcal{A}} t\sigma$

- $\phi \equiv \top$
- ϕ *est de la forme* $\phi_1 \wedge \phi_2$ *et* $\sigma \in \mathcal{S}(\mathcal{A}, \phi_1) \cap \mathcal{S}(\mathcal{A}, \phi_2)$
- ϕ *est de la forme* $\phi_1 \vee \phi_2$ *et* $\sigma \in \mathcal{S}(\mathcal{A}, \phi_1) \cup \mathcal{S}(\mathcal{A}, \phi_2)$
- ϕ *est de la forme* $\neg\phi_1$ *et* $\sigma \notin \mathcal{S}(\mathcal{A}, \phi_1)$
- ϕ *est de la forme* $\exists x : \phi_1$ *et il existe une* $\mathcal{A}$*-substitution* θ *de domaine* $\{x\}$ *telle que* $\theta\sigma \in \mathcal{S}(\mathcal{A}, \phi_1)$
- ϕ *est de la forme* $\forall x : \phi_1$ *et pour toute* $\mathcal{A}$*-substitution* θ *de domaine* $\{x\}$, $\theta\sigma \in \mathcal{S}(\mathcal{A}, \phi_1)$.

On notera $\mathcal{A} \models \phi$ si, pour toute $\mathcal{A}$-substitution σ de domaine $VL(\phi)$, $(\mathcal{A}, \sigma) \models \phi$. On dira alors que ϕ est *valide* dans $\mathcal{A}$.

Deux formules équationnelles ϕ_1 et ϕ_2 sont (sémantiquement) *équivalentes* si elles ont mêmes modèles. On notera dans ce cas $\phi_1 \sim \phi_2$. Certaines équivalences sont bien connues. Rappelons en quelques-unes (sans chercher à être exhaustif pour l'instant):

- $\vee$ est associatif et commutatif, $\bot$ en est un élément neutre et $\top$ est un élément absorbant. Enfin, $\vee$ est idempotent et distributif par rapport à $\wedge$.
- $\exists x : (\exists y : \phi) \sim \exists y : (\exists x : \phi)$ et $\exists x : (\exists x : \phi) \sim \exists x : \phi$. Si bien que nous écrirons $\exists x_1, \ldots, x_n : \phi$, les variables $x_1, \ldots, x_n$ formant un ensemble, au lieu de $\exists x_1 : (\exists \ldots P) \ldots)$. La même propriété vaut pour le quantificateur $\forall$.
- $\neg(\neg\phi) \sim \phi$, $\neg(\phi_1 \vee \phi_2) \sim \neg\phi_1 \wedge \neg\phi_2$ et $\neg(\exists x_1, \ldots x_n : \phi) \sim \forall x_1, \ldots, x_n : \neg\phi$.
- $(\forall x : \phi_1) \wedge (\forall y : \phi_2) \sim \forall x, y : (\phi_1 \wedge \phi_2)$ et

$$(\forall x : \phi_1) \vee (\forall y : \phi_2) \sim \forall x', y' : (\phi_1\{x \to x'\} \vee \phi_2\{y \to y'\})$$

  où x' et y' sont des variables distinctes de même sorte que x et y respectivement et qui ne sont pas dans $Var(\phi_1, \phi_2)$. $\phi\{x \to x'\}$ désigne la formule ϕ dans laquelle toute occurrence de x a été remplacée par x'.

On peut aussi noter les propriétés de "monotonie" :

Lemme 2.5 *Soient* ϕ_1 *et* ϕ_2 *deux formules équationnelles et* $\mathcal{A}$ *une* F*-algèbre telles que* $\mathcal{S}(\mathcal{A}, \phi_1) \subseteq \mathcal{S}(\mathcal{A}, \phi_2)$. *Alors, pour toute formule équationnelle* ϕ_3 *et pour toute variable* x *on a les inclusions suivantes :*

$$\begin{array}{rcl}
\mathcal{S}(\mathcal{A}, \phi_1 \vee \phi_3) & \subseteq & \mathcal{S}(\mathcal{A}, \phi_2 \vee \phi_3) \\
\mathcal{S}(\mathcal{A}, \phi_1 \wedge \phi_3) & \subseteq & \mathcal{S}(\mathcal{A}, \phi_2 \wedge \phi_3) \\
\mathcal{S}(\mathcal{A}, \neg\phi_2) & \subseteq & \mathcal{S}(\mathcal{A}, \neg\phi_1) \\
\mathcal{S}(\mathcal{A}, \exists x : \phi_1) & \subseteq & \mathcal{S}(\mathcal{A}, \exists x : \phi_2) \\
\mathcal{S}(\mathcal{A}, \forall x : \phi_1) & \subseteq & \mathcal{S}(\mathcal{A}, \forall x : \phi_2)
\end{array}$$

Enfin, il est bien connu qu'il existe un algorithme permettant de transformer une formule quelconque en une formule en *forme prénexe* [Gal86,Caf86]. c'est-à-dire une formule dans laquelle tous les quantificateurs sont en tête et toutes les négations sont à l'occurrence

la plus interne. Ainsi, le symbole $\neg$ n'apparait plus dans une formule équationnelle en forme prénexe puisque l'on note $s \neq t$ pour $\neg(s = t)$. Une forme prénexe d'une formule équationnelle est ainsi une suite de quantificateurs suivie d'une formule booléenne dont les atomes sont des équations ou des diséquations.

2.3 Problèmes équationnels

2.3.1 Résolution de formules équationnelles

Nous nous intéressons dans la suite à *résoudre* des formules équationnelles. Résoudre une formule équationnelle ϕ dans la F-algèbre $\mathcal{A}$ (resp. dans la variété de F-algèbres $\mathcal{V}$) et par rapport à l'ensemble fini $\mathcal{I}$ d'inconnues principales supposé contenir les variables libres de ϕ c'est trouver toutes les $\mathcal{A}$-substitutions σ (resp. toutes les substitutions σ) telles que $Dom(\sigma) = \mathcal{I}$ et que $(\mathcal{A}, \sigma) \models \phi$ (resp. pour toute F-algèbre $\mathcal{A}$ de $\mathcal{V}$, $\mathcal{A} \models \phi\sigma$).

Par exemple,

- Lorsque ϕ est une équation $s = t$, $\mathcal{A} = T(F, X)/ =_E$ et $\mathcal{I} = Var(\phi)$, résoudre ϕ, c'est *unifier* s et t dans la théorie équationnelle définie par E. De façon plus générale, lorsque ϕ ne contient ni quantificateur ni négation on dit que ϕ est un *problème d'unification* .

  De tels problèmes on été largement étudiés (et continuent de l'être). Citons par exemple [Her30,MM82,Sie84,Kir85,Uni87,Uni88].

- Lorsque ϕ est une conjonction d'équations dont toutes les variables sont quantifiées universellement, $\mathcal{I} = \emptyset$ et $\mathcal{A} = T(F, X)/ =_E$, la résolution de ϕ est un *problème du mot.*

  Ces problèmes ont eux aussi fait l'objet de nombreuses études. Citons [KB70,Hue81, JK86a,BDH86,HR87], entre autres.

- lorsque $\mathcal{A} = T(F)/ =_E$ et $\mathcal{I} = \emptyset$, la résolution de ϕ est un problème de preuve par induction dans une théorie équationnelle. .

  Là encore, les références abondent. Entre autres travaux récents citons [Mus80, HH82,JK86b,Bac88].

On voit que le problème que nous nous posons est très général et recouvre plusieurs problèmes bien connus.

2.3.2 Problèmes équationnels : syntaxe

Pour résoudre les formules équationnelles nous pouvons déjà nous limiter aux formules en forme prénexe. Nous allons de plus considérer des formules équationnelles particulières: les problèmes équationnels. En effet, notre méthode de résolution, basée sur une "élimination des quantificateurs" ne fait intervenir que les deux quantificateurs "les plus internes" de la formule. Les problèmes équationnels n'auront donc que deux quantificateurs.

Définition 2.6 *Un* problème équationnel *est une formule équationnelle $\mathcal{P}$ en forme prénexe*

$$\mathcal{P} \equiv \exists w_1, \ldots w_n, \forall y_1, \ldots, y_m : P$$

P ne contenant pas de quantificateurs, et telle que $\{w_1, \ldots, w_n\} \cap \{y_1, \ldots, y_m\} = \emptyset$.

Dans un problème équationnel $\mathcal{P}$ on distingue ainsi trois sortes de variables :

- les *paramètres* (notés $y, y', y_1, \ldots$) qui sont les variables quantifiées universellement dans $\mathcal{P}$
- les *inconnues auxiliaires* (notés $w, w', w_n, \ldots$) qui sont les variables quantifiées existentiellement dans $\mathcal{P}$. Intuitivement, ce sont des inconnues dont on peut avoir besoin, soit temporairement, soit pour exprimer les solutions.
- les *inconnues principales* dont font partie les variables libres de $\mathcal{P}$ (avec comme éléments typiques $x, x', x_i, \ldots$). Intuitivement, ce sont les "inconnues initiales" du problème, celles par rapport auxquelles on souhaite le "résoudre".

Si $\mathcal{P}$ est un problème équationnel, on notera $Param(\mathcal{P})$ l'ensemble de ses paramètres et $Inc(\mathcal{P})$ (Inc pour "inconnues") l'ensemble $Var(\mathcal{P}) - Param(\mathcal{P})$[7].

En fait, on peut voir l'ensemble des problèmes équationnels (F étant donné) comme une algèbre de termes quotientée par les relations d'algèbre booléenne. Ces relations font partie de celles qui sont induites par $\sim$ sur le sous-ensemble des problèmes équationnels. La figure 2.1 donne une présentation à la OBJ [FGJM85] (avec sous sortes) de l'algèbre des problèmes équationnels. Nous supposons que les objets de sorte *variable* et *terme* sont déjà définis. Pour simplifier, nous supposons aussi que S (ensemble des sortes) ne contient qu'un élément. Enfin, dans cette présentation, les axiomes (équationnels) de l'algèbre sont notés au moyen du signe == afin d'éviter les confusions avec les autres signes d'égalité déjà employés.

De telles définitions permettent de donner un sens aux substitutions et aux remplacements au sein de problèmes équationnels. De même cela permet d'avoir une notation rigoureuse pour la transformation des problèmes, comme nous allons le voir. Enfin, comme $\neg$ n'apparait pas dans un problème équationnel, il est possible d'orienter les équations de la figure 2.1 en un système de réécriture canonique modulo les axiomes d'associativité et de commutativité. On peut, en fait, obtenir plusieurs systèmes canoniques. En particulier, selon la façon d'orienter les axiomes de distributivité, on obtient soit des formes normales conjonctives soit des formes normales disjonctives. (Ce qui est bien connu en logique).

2.3.3 Problèmes équationnels : sémantique

Définition 2.7 *Soient $\mathcal{P}$ un problème équationnel, $\mathcal{I}$ un ensemble fini de variables contenant les variables libres de $\mathcal{P}$ et $\mathcal{A}$ une F-algèbre telle que $\mathcal{I} \cap \mathcal{A} = \emptyset$. Une $\mathcal{A}$-substitution σ est une* $\mathcal{A}$-solution *de $\mathcal{P}$ (par rapport à $\mathcal{I}$) si $Dom(\sigma) = \mathcal{I}$ et $\sigma \in \mathcal{S}(\mathcal{A}, \mathcal{P})$.*

[7] Certaines inconnues de $\mathcal{I}$ peuvent ne pas figurer dans $\mathcal{P}$ et donc n'être pas contenues dans $Inc(\mathcal{P})$.

OBJ EQUATIONAL_PROBLEMS

SORTS *terme variable ens_var systeme ep*

SUBSORTS $\begin{array}{rcl} \textit{système} & < & \textit{ep} \\ \textit{var} & < & \textit{ens\_var} \end{array}$

IMPORT *terme, variable*

OPERATORS

$$\begin{array}{rcl} \_=\_,\_\neq\_ & : & \textit{terme} \times \textit{terme} \rightarrow \textit{système} \\ \_\vee\_,\_\wedge\_ & : & \textit{système} \times \textit{système} \rightarrow \textit{système} \\ \top,\bot & : & \rightarrow \textit{système} \\ \_,\_ & : & \textit{var} \times \textit{ens\_var} \rightarrow \textit{ens\_var} \\ \forall\_,\_ & : & \textit{ens\_var} \times \textit{système} \rightarrow \textit{ep} \\ \exists\_,\_ & : & \textit{ens\_var} \times \textit{ep} \rightarrow \textit{ep} \end{array}$$

VARIABLES

$$\begin{array}{rcl} s,t & : & \textit{terme} \\ P,P_1,P_2,P_3,Q & : & \textit{système} \\ \vec{z},\vec{z'},\vec{z''} & : & \textit{ens\_var} \end{array}$$

EQUATIONS

$$\begin{array}{rcll} s=t & == & t=s & \text{commutativité de } = \text{ et } \neq \\ s\neq t & == & t\neq s & \\ P\vee Q & == & Q\vee P & \text{commutativité de } \vee \text{ et } \wedge \\ P\wedge Q & == & Q\wedge P & \\ P_1\wedge(P_2\wedge P_3) & == & (P_1\wedge P_2)\wedge P_3 & \text{associativité} \\ P_1\vee(P_2\vee P_3) & == & (P_1\vee P_2)\vee P_3 & \\ P_1\wedge(P_2\vee P_3) & == & (P_1\wedge P_2)\vee(P_1\wedge P_3) & \text{distributivité} \\ P_1\vee(P_2\wedge P_3) & == & (P_1\vee P_2)\wedge(P_1\vee P_3) & \\ P\wedge P & == & P & \text{idempotence} \\ P\vee P & == & P & \\ P\wedge\bot & == & \bot & \text{éléments absorbants} \\ P\vee\top & == & \top & \\ P\wedge\top & == & P & \text{éléments neutres} \\ P\vee\bot & == & P & \\ \vec{z},\vec{z'} & == & \vec{z'},\vec{z} & \text{propriétés des ensembles} \\ \vec{z},(\vec{z'},\vec{z''}) & == & (\vec{z},\vec{z'}),\vec{z''} & \\ \vec{z},\vec{z} & == & \vec{z} & \\ \exists\vec{z},\exists\vec{z'} & == & \exists\vec{z}.\vec{z'} & \end{array}$$

Figure 2.1: Algèbre des problèmes équationnels

On peut remarquer que cette définition est indépendante du représentant choisi dans l'algèbre des problèmes équationnels puisque les équations de cette algèbre préservent les modèles.

Nous noterons $\mathcal{S}(\mathcal{A},\mathcal{P},\mathcal{I})$ l'ensemble des $\mathcal{A}$-solutions de $\mathcal{P}$ par rapport à $\mathcal{I}$. Lorsque $\mathcal{S}(\mathcal{A},\mathcal{P},\mathcal{I}) = \mathcal{S}(\mathcal{A},\mathcal{P}',\mathcal{I})$, on dira que $\mathcal{P}$ et $\mathcal{P}'$ sont *équivalents* par rapport à $\mathcal{A}$ et $\mathcal{I}$[8], ce que nous noterons $\mathcal{P} \approx_{\mathcal{A},\mathcal{I}} \mathcal{P}'$ ou simplement $\mathcal{P} \approx \mathcal{P}'$ s'il n'y a pas d'ambiguité.

Si $A \subseteq \mathcal{A}$, une *solution dans* A du problème $\mathcal{P}$ est une $\mathcal{A}$-solution σ de $\mathcal{P}$ telle que $\sigma(Dom(\sigma)) \subseteq A$.

2.4 Exemples

Nous nous intéresserons dans la suite à certaines F-algèbres particulières:

- $T(F)$ et $T(F,X)$
- Les quotients de $T(F,X)$ par une congruence $=_E$ engendrée par un ensemble fini d'équations
- $RT(F)$, l'algèbre des arbres rationnels
- $NF_{\mathcal{R}}$ l'algèbre des termes de $T(F)$ qui sont irréductibles pour un système de réécriture $\mathcal{R}$.

Donnons un exemple dans chaque cas.

2.4.1 Exemples dans $T(F)$ et $T(F,X)$

Exemple 2.1 Cet exemple très simple illustre le fait que les problèmes d'unification sont des problèmes équationnels.
$S = \{\underline{s}\}$, $F = \{a : \rightarrow \underline{s};\ g : \underline{s} \rightarrow \underline{s};\ f : \underline{s} \times \underline{s} \rightarrow \underline{s}\}$, $\mathcal{P} \equiv f(x,g(x')) = f(g(a),x)$ est un problème équationnel sans paramètre et sans inconnue auxiliaire dont les inconnues principales sont x et x'. C'est un problème d'unification dont les solutions dans $T(F,X)$ sont représentées par le système $x = g(a) \wedge x' = a$. La seule solution dans $T(F,X)$ (comme dans $T(F)$) est alors $\{x \rightarrow g(a);\ x' \rightarrow a\}$.

Exemple 2.2 L'exemple ci-dessous est lié au problème de complétude suffisante déjà évoqué plus haut:

$$F = \{0 : \rightarrow int;\ p, s : int \rightarrow int;\ eq : int \times int \rightarrow bool;\ true, false : \rightarrow bool\}$$

$$\begin{aligned}\mathcal{P} \equiv\ & \forall y_1, y_2 :\ eq(x_1,x_2) \neq eq(y_1,y_1) \wedge eq(x_1,x_2) \neq eq(s(y_1),s(y_2)) \\ & \wedge\ eq(x_1,x_2) \neq eq(p(y_1),p(y_2)) \wedge eq(x_1,x_2) \neq eq(y_1,s(y_1)) \\ & \wedge\ eq(x_1,x_2) \neq eq(s(y_1),y_1) \wedge eq(x_1,x_2) \neq eq(y_1,p(y_1)) \\ & \wedge\ eq(x_1,x_2) \neq eq(p(y_1),y_1)\end{aligned}$$

[8] Pour faire le lien avec la notion précédente d'équivalence, deux problèmes $\mathcal{P}$ et $\mathcal{P}'$ ayant même ensemble de variables libres $\mathcal{I}$ sont équivalents ssi pour toute F-algèbre $\mathcal{A}$, ils sont équivalents par rapport à $\mathcal{A},\mathcal{I}$.

Le problème $\mathcal{P}$ a pour inconnues principales x_1 et x_2 et pour paramètres y_1 et y_2 (il n'y a pas d'inconnue auxiliaire). Il n'a pas de solution dans $T(F)$ si et seulement si le système de réécriture

$$\begin{array}{rcl} eq(y,y) & \rightarrow & true \\ eq(s(x),s(y)) & \rightarrow & eq(x,y) \\ eq(p(x),p(y)) & \rightarrow & eq(x,y) \end{array} \qquad \begin{array}{rcl} eq(y,s(y)) & \rightarrow & false \\ eq(p(x),x) & \rightarrow & false \end{array} \qquad \begin{array}{rcl} eq(s(y),y) & \rightarrow & false \\ eq(x,p(x)) & \rightarrow & false \end{array}$$

définit complètement eq.

Notons que ce n'est pas le cas ici[9] car les $T(F)$-substitutions $\{x_1 \rightarrow p(0);\ x_2 \rightarrow s(0)\}, \ldots, \{x_1 \rightarrow p^n(0);\ x_2 \rightarrow s^m(0)\}, \ldots, \{x_1 \rightarrow 0;\ x_2 \rightarrow s(s(0))\}, \ldots, \{x_1 \rightarrow s^k(0);\ x_2 \rightarrow s^{k+k'+2}(0)\}, \ldots$ sont (entre autres) des solutions de $\mathcal{P}$ puisque, pour toute substitution fermée sur y_1, y_2, chacune des diséquations du problème est satisfaite.

On peut également noter que, supposant que $\mathcal{R}$ contient en outre les deux règles $p(s(x)) \rightarrow x$ et $s(p(x)) \rightarrow x$ ces solutions sont irréductibles pour $\mathcal{R}$. [10]

2.4.2 Exemples dans $T(F,X)/=_E$ et $T(F)/=_E$

Exemple 2.3 Cet exemple illustre un *problème du mot* . C'est le problème bien connu de la théorie des groupes définie "minimalement". $S = \{\underline{s}\}$, $F = \{e : \rightarrow \underline{s};\ I : \underline{s} \rightarrow \underline{s};\ * : \underline{s} \times \underline{s} \rightarrow \underline{s}\}$ et

$$E = \{ \quad \begin{array}{rcl} x * e & == & x \\ x * I(x) & == & e \\ x * (y * z) & == & (x * y) * z \end{array} \quad \}$$

On s'intéresse à la résolution du problème :

$$\forall y_1, y_2 : I(y_1 * y_2) = I(y_2) * I(y_1)$$

dans $T(F,X)/=_E$.

Ce problème n'a pas d'inconnue; toutes ses variables sont des paramètres. Résoudre un tel problème dans $T(F,X)$ c'est décider si $I(y_1 * y_2) = I(y_2) * I(y_1)$ est (ou non) un théorème de la variété équationnelle définie par E.

On peut démontrer que ce problème est équivalent à $\top$ (voir [HO80] par exemple) en orientant les équations de E puis en "complétant" [HO80,BDH86]. Le système de réécriture ainsi obtenu en un système de réécriture canonique $\mathcal{R}$. Les deux membres de l'équation se réduisent alors en un même terme si et seulement si l'égalité est un théorème de la variété.

[9] En fait il n'est pas possible d'obtenir une spécification complète de eq sur les entiers relatifs avec seulement un nombre fini d'axiomes équationnels et pas de fonction cachée comme nous le verrons dans le chapitre 6.

[10] Nous y reviendrons dans les exemples de solutions dans NF.

Exemple 2.4 Prenons ici un exemple moins classique faisant intervenir des diséquations. $S = \{\underline{s}\}$, $F = \{0, 1 : \rightarrow \underline{s};\ + : \underline{s} \times \underline{s} \rightarrow \underline{s}\}$.$E$ est l'ensemble d'axiomes équationnels:

$$\begin{array}{rcl} 0 + x & == & x \\ x + y & == & y + x \\ x + (y + z) & == & (x + y) + z \end{array}$$

Cet ensemble d'axiomes établit la commutativité et l'associativité de $+$ ainsi que l'existence d'un élément neutre.

Intéressons nous alors au problème suivant:

$$\begin{array}{rcl} \mathcal{P} & \equiv & \forall y_1, y_2 :\ (x \neq (x' + y_1) + 1) \\ & & \wedge\ (x + x' \neq (y_1 + (y_1 + (y_2 + 1))) \ \vee\ (x + (y_1 + y_2) \neq x' + (1 + 1))) \\ & & \wedge\ (x' \neq x + (y_1 + 1) \ \vee\ (x \neq y_1 + y_2)) \end{array}$$

Ce problème a pour inconnues x' et x et pour paramètres y_1 et y_2. Il n'a aucune signification précise, mais est tiré indirectement d'un problème de complétude suffisante.

$\mathcal{P}$ possède deux ensembles de $T(F)/=_E$ solutions qui sont décrits par :

1. $x = x' = 0$
2. $x = 1$ et $x' = 1 + z_1$

Autrement dit $\mathcal{P}$ a même ensemble de $T(F)/=_E$-solutions que la formule équationnelle:

$$(x = 0 \wedge x' = 0) \ \vee\ (\exists z_1 :\ x = 1 \wedge x' = 1 + z_1)$$

On voit sur cet exemple l'utilisation d'une *variable auxiliaire* z_1 qui permet d'exprimer l'ensemble des solutions. L'introduction de telles variables est d'un usage courant dans les problèmes d'unification dans les théories équationnelles. Elle est nécessaire, par exemple, pour l'unification modulo l'associativité-commutativité [Sti81].

2.4.3 Exemple dans les arbres rationnels

Les arbres infinis (et en particulier les arbres rationnels) jouent un rôle important en informatique, par exemple dans la sémantique des programmes récursifs. Ils interviennent aussi dans les problèmes d'unification lorsqu'on évite le test d'occurrence [Hue76].

Rappelons la définition des arbres rationnels.

L'ensemble des positions d'un arbre infini étiqueté est défini comme l'ensemble des positions d' un arbre fini, excepté qu'il peut être infini. La fonction d'étiquetage doit vérifier, elle aussi, des propriétés analogues; si $t(u) = f$ et $f : \underline{s}_1 \times \ldots \times \underline{s}_n \rightarrow \underline{s}$ alors

- $u \cdot i \in Pos(t)$ ssi $1 \leq i \leq n$
- $sort(t/u) = \underline{s}$ et $sort(t/u \cdot i) = \underline{s}_i$.

Un arbre infini est *rationnel* s'il n'a qu'un nombre fini de sous-arbres distincts. La F-algèbre des arbres rationnels est notée $RT(F)$. Ce n'est une pas une algèbre *localement libre* [Mal71], au contraire de $T(F)$.

On possède un certain nombre de caractérisations des arbres rationnels [Hue76,Cou81]. Nous ne retiendrons que la caractérisation en termes d'équations:

Théoreme 2.8 [Hue76,Cou81] *Un système d'équations $x_1 = t_1 \wedge \ldots \wedge x_n = t_n$ dans lequel $x_1, \ldots x_n$ sont des variables distinctes, $Var(t_1, \ldots, t_n) = \{x_1, \ldots, x_n\}$ et, pour tout i, $x_i \not\equiv t_i$, possède une unique $RT(F)$-solution par rapport à $\{x_1, \ldots, x_n\}$.*

A. Colmerauer [Col82,Col84] a étudié les systèmes d'équations et de diséquations dans les arbres rationnels, car de tels systèmes permettent d'exprimer une partie du contrôle en PROLOG II. Bien sûr, les systèmes d'équations et de diséquations sont des problèmes équationnels.

Exemple 2.5 $S = \{\underline{s}\}$, $F = \{0 : \rightarrow \underline{s};\ g : \underline{s} \times \underline{s} \rightarrow \underline{s}\}$ et

$$\mathcal{P} \equiv \forall y_1, y_2 : y_1 \neq g(y_1, x) \vee y_1 \neq g(y_1, y_2) \vee y_2 \neq 0$$

Si l'on considère ce problème dans $T(F)$, n'importe quelle $T(F)$-substitution est une $T(F)$-solution puisque, dans les arbres finis, la diséquation $y_1 \neq g(y_1, x)$ est toujours satisfaite. Il n'en est pas de même dans les arbres infinis. En fait, toute $RT(F)$-substitution est une $RT(F)$-solution excepté $\{x \rightarrow 0\}$. En effet, dans ce dernier cas, on peut infirmer les trois diséquations en choisissant $y_1 = g(y_1, 0)$ et $y_2 = 0$.

2.4.4 Exemples dans NF

A nouveau, revenons à un exemple de complétude suffisante. Comme déjà dit plus haut, ce problème se ramène dans un cas simple à l'absence de solution dans $T(C)$ pour un certain problème équationnel. De façon plus générale, ce problème se ramène à l'absence de solution dans NF, ensemble des termes fermés irréductibles pour un système de réécriture[11].

Exemple 2.6 Reprenons tout d'abord l'exemple 2.2. Si l'on ajoute les règles $p(s(x)) \rightarrow x$ et $s(p(x)) \rightarrow x$ qui correspondent à la définition des entiers relatifs, le système de réécriture reste canonique. *eq* est alors complètement défini ssi il n'existe pas de solution dans NF au problème $\mathcal{P}$ de l'exemple 2.2. Il se trouve que les solutions que nous avions exhibées alors sont dans NF, ce qui prouve que *eq* n'est pas complètement défini, même avec ces nouvelles relations.

[11] Rappelons que, comme vu plus haut, NF est muni "canoniquement" d'une structure de F-algèbre

Exemple 2.7 Donnons ici un autre exemple très simple.
$S = \{\underline{s}\}$, $F = \{0 : \rightarrow \underline{s};\ s : \underline{s} \rightarrow \underline{s};\ + : \underline{s} \times \underline{s} \rightarrow \underline{s}\}$ et $\mathcal{R}$ est le système:

$$\begin{array}{rcl} s(s(0)) & \rightarrow & 0 \\ x + x & \rightarrow & 0 \\ 0 + x & \rightarrow & x \\ x + 0 & \rightarrow & x \end{array}$$

On s'intéresse à la complétude de la définition de $+$. Il n'est pas possible ici de séparer F en deux sous-ensembles C et D sans qu'il y ait de relation entre termes de $T(C)$. Mais on peut encore ramener la complète définition de $+$ à l'existence de solutions dans NF pour le problème:

$$\mathcal{P} \equiv \forall y : x_1 + x_2 \neq y + y \wedge x_1 + x_2 \neq 0 + y \wedge x_1 + x_2 \neq y + 0$$

Pour résoudre cette question, on commence par chercher les solutions dans $T(\{0, s\})$, puis on cherche parmi les solutions proposées celles qui sont irréductibles [Com86]. Il se trouve ici que les solutions dans $T(\{0, s\})$ sont de la forme $\{x_1 \rightarrow s^k(0);\ x_2 \rightarrow s^m(0)\}$ avec $k, m \geq 1$ et $k \neq m$. Or tout terme $s^k(0)$ avec $k \geq 2$ est réductible par la première règle du système, ce qui prouve qu'il n'y a pas de solution dans NF. Par conséquent, $+$ est complètement défini.

2.5 Travaux voisins

On trouve dans certains travaux ([Mal71,Kun87,Mah88a]) une expression différente des problèmes que nous nous posons.

Un ensemble (récursif) $\mathcal{F}$ de formules équationnelles sans variables libres (aussi appelées *phrases*) est une *axiomatisation complète* de la classe d'algèbres $\mathcal{K}$ ssi les algèbres de $\mathcal{K}$ sont les modèles de $\mathcal{F}$. Autrement dit, si $\vdash$ est la relation de déduction associée à un système complet dans le CP1, $\mathcal{F}$ est une axiomatisation complète de $\mathcal{K}$ ssi

$$\mathcal{K} \models \phi \Leftrightarrow \mathcal{F} \vdash \phi$$

Mal'cev prouve que les axiomes :

(D)	$f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n)$	$\Rightarrow$	$t_1 = u_1 \wedge \ldots \wedge t_n = u_n$
(I)	$f(t_1, \ldots, t_n) \neq g(u_1, \ldots, u_m)$		Si f≠g
(O)	$t[x] \neq x$		Si t est un terme quelconque distinct de x

constituent une axiomatisation complète de la classe des algèbres *localement libres.* [12] Une F-algèbre $\mathcal{A}$ est dite localement libre lorsque toute sous algèbre de $\mathcal{A}$ qui est finiment engendrée possède un ensemble (éventuellement infini) de générateurs libres. Les algèbres libres (telles que $T(F)$ ou $T(F, X)$) sont localement libres [13].

[12] Cette axiomatisation n'est pas finie puisque les axiomes (O) sont en nombre infini.

[13] Mais il existe des algèbres localement libres qui ne sont pas libres comme $\mathbf{Z}$, ensemble des entiers relatifs considéré comme $\{0, succ\}$-algèbre.

Avec cette terminologie, les résultats que nous donnons dans le chapitre suivant permettent de prouver (comme dans [Mah88a]) que les axiomes $(D), (I), (O)$ plus l'axiome:

$$(DCA) \quad \forall x, \bigvee_{f \in F} \exists \vec{z}, x = f(\vec{z})$$

constituent une axiomatisation complète de $T(F)$. (Ce qui ne peut être déduit des résultats de Mal'cev puisqu'il y a "plus de théorèmes" dans $T(F)$ que dans la classe des algèbres libres). La façon dont nous abordons cette preuve permettra aussi, comme nous le verrons, d'effectuer certaines généralisations (par exemple au cas équationnel et aux algèbres avec sortes ordonnées).

Plusieurs travaux récents ont aussi abordé des problèmes voisins. L'introduction de systèmes d'équations et de diséquations pour résoudre les problèmes de complétude suffisante a constitué, de façon plus ou moins explicite, l'idée de base des travaux présentés dans [Thi84,Com86,Kuc88]. L'étude des problèmes équationnels dans leur généralité avec application à la complétude suffisante est lancée dans [KL87]. Mais, d'une part les problèmes introduits dans [KL87] ne contiennent pas de variables libres, ce qui empêche d'exprimer correctement la résolution de tels problèmes, d'autre part il manquait une description de ce qu'est le résultat d'une telle transformation. Le problème de la résolution des problèmes équationnels dans le cadre présenté ici a été ensuite résolu dans [CL88]. Enfin, il faut aussi citer le travail de HJ. Bürckert [Bur88] où est abordé le problème de la résolution de certains problèmes équationnels dans $T(F, X)/=_E$. Mais, à la différence de notre travail, les problèmes considérés dans cet article ne comportent pas de variable quantifiée universellement.

Chapitre 3

Transformation des problèmes équationnels

Dans ce chapitre, nous montrons comment transformer les problèmes équationnels tout en ne modifiant pas l'ensemble des solutions.

Dans un premier temps, nous nous intéressons aux règles qui ont cette propriété, sans nous occuper ni de terminaison, ni des propriétés éventuelles des problèmes en forme irréductible. Par contre, les règles sont énoncées sous une forme aussi générale que possible, les résultats de correction étant eux aussi énoncés avec des hypothèses aussi faibles que possible.

Dans un deuxième temps, nous nous intéressons à l'utilisation de ces règles pour obtenir des "formes résolues". Il nous faut alors considérer des représentants de problèmes équationnels et introduire un *contrôle* dans l'utilisation de celles-ci. Après avoir précisé ces notions dans un cadre général, nous donnons (sections 3.4,3.6,3.7) un contrôle (le plus libéral possible) permettant d'une part de prouver la terminaison des transformations, d'autre part d'assurer de "bonnes" propriétés des formes irréductibles.

Une conséquence de ces résultats est la décidabilité de la validité dans $T(F)$ des formules du premier ordre dont le seul symbole de prédicat est $=$.

Une partie du travail présenté dans ce chapitre a été mené en collaboration avec P. Lescanne.

3.1 Règles de transformation

3.1.1 Généralités

Une *règle de transformation* est un ensemble de (shémas de) règles de réécriture dans l'algèbre des problèmes équationnels. La plupart du temps il s'agira *d'une* règle de réécriture. Mais parfois, il pourra être utile d'utiliser la même notation pour désigner plusieurs règles. Par exemple, une règle comme:

$$f(t_1,\ldots,t_n) = g(u_1,\ldots,u_m) \mapsto \bot \quad Si\ f \neq g$$

dénote l'ensemble des règles $\{f(t_1,\ldots,t_n) = g(u_1,\ldots,u_m) \mapsto \bot \mid f,g \in F, f \neq g\}$.

De même, dans certains cas, il peut être utile d'avoir en membre droit non pas un terme mais une fonction appliquée à un terme. On pourrait toujours éviter ce genre de notation en définissant complètement les opérations nécessaires dans l'algèbre des problèmes équationnels. Mais de telles définitions sont fastidieuses et n'apportent rien. Dans tous les cas, l'ensemble de règles de réécriture dénoté par une de nos règles de transformation apparaitra clairement.

De cette façon, nous pourrons parler de terminaison et de confluence d'un système de règles de transformation, comme de la terminaison et de la confluence de l'ensemble de règles de réécriture qu'elles dénotent. Bien sûr, à un ensemble de règles de transformation $\mathcal{R}$ est associée une relation de réduction $\mapsto_{\mathcal{R}}$.

Définition 3.1 *Etant donné une F-algèbre $\mathcal{A}$ et un ensemble fini de variables $\mathcal{I}$, un ensemble de règles de transformation $\mathcal{R}$ est dit* correct *par rapport à $\mathcal{A},\mathcal{I}$ si, pour tout problème équationnel $\mathcal{P}$,*

$$(\mathcal{P} \mapsto_{\mathcal{R}} \mathcal{P}') \;\Rightarrow\; (\mathcal{S}(\mathcal{A},\mathcal{P}',\mathcal{I}) \subseteq \mathcal{S}(\mathcal{A},\mathcal{P},\mathcal{I}))$$

Définition 3.2 *Etant donné une F-algèbre $\mathcal{A}$ et un ensemble fini de variables $\mathcal{I}$, un ensemble de règles de transformation $\mathcal{R}$ est dit* conservatif *(resp.* globalement conservatif*) par rapport à $\mathcal{A},\mathcal{I}$ si, pour tout problème équationnel $\mathcal{P}$,*

$$(\mathcal{P} \mapsto_{\mathcal{R}} \mathcal{P}') \;\Rightarrow\; (\mathcal{S}(\mathcal{A},\mathcal{P},\mathcal{I}) \subseteq \mathcal{S}(\mathcal{A},\mathcal{P}',\mathcal{I}))$$

(resp.

$$\mathcal{S}(\mathcal{A},\mathcal{P},\mathcal{I}) \subseteq \bigcup_{\mathcal{P}',\mathcal{P}\mapsto_{\mathcal{R}}\mathcal{P}'} \mathcal{S}(\mathcal{A},\mathcal{P},\mathcal{I})$$

)

La notion de "globale conservation" est introduite pour permettre de tirer parti du non déterminisme de l'application des règles : une règle telle que :

$$(Ex_1) \quad P \wedge x \neq u \;\mapsto\; \bigvee_{f\in C}(\exists z_1,\ldots,z_m :\; (P \wedge x \neq u) \wedge x = f(z_1,\ldots z_m))$$

ne transforme en effet pas un problème équationnel en un autre problème équationnel mais en une disjonction de problèmes. Nous l'utiliserons donc sous la forme:

$$(Ex_1) \quad P \wedge x \neq u \;\mapsto\; \exists z_1,\ldots,z_m :\; (P \wedge x \neq u) \wedge x = f(z_1,\ldots z_m) \qquad Si\ f \in F$$

Mais cette dernière règle n'est plus que *globalement* conservative.

Définition 3.3 *Lorsque $\mathcal{R}$ est à la fois correct et conservatif (resp. globalement conservatif) par rapport à $\mathcal{A},\mathcal{I}$, on dit que $\mathcal{R}$ est* fortement adéquat *(resp.* adéquat*) par rapport à $\mathcal{A},\mathcal{I}$.*

Lorsqu'un ensemble de règles est fortement adéquat, il suffit de calculer un seul "successeur" de $\mathcal{P}$. Remarquons aussi qu'un ensemble de règles est fortement adéquat si et seulement si chaque règle de cet ensemble est fortement adéquate. Enfin, un ensemble de règles fortement adéquat est évidemment adéquat.

3.1.2 Énoncé des règles

Les résultats de correction et d'adéquation des règles seront donnés dans le paragraphe suivant. Nous nous contentons ici d'énoncer ces règles de façon à les obtenir groupées et à pouvoir s'y référer dans la suite.

Nous utiliserons les conventions suivantes dans la description des règles de transformation:

- toute chaine commençant par s, t, u, v désignera une variable de sorte *terme*
- toute chaine commençant par w, x, y, z désignera une variable de sorte *variable*
- toute chaine commençant par P, Q, R désignera une variable de sorte *système* (c'est-à-dire un problème équationnel sans quantificateur)
- les vecteurs $(\vec{x}, \vec{y}, \vec{z}, \ldots)$ désigneront des variables de sorte *seq_var*
- $\mathcal{P}$ désignera une variable de sorte *ep* (i.e "problème équationnel")

D'autre part nous nous efforcerons d'adopter la convention lexicographique indiquée dans le chapitre précédent concernant les variables : les paramètres sont désignés par les variables $y, y_1, y', \ldots$, les inconnues principales par les variables $x, x', x_1, \ldots$, les inconnues auxiliaires par les variables $w, w', w_1, \ldots$ et les variables quelconques par $z, z', z_1, \ldots$

La figure 3.1 énonce les règles dont la correction ne dépend pas de l'algèbre $\mathcal{A}$ considérée. On y trouve des règles très classiques comme le remplacement, la fusion et l'élimination des équations triviales, avec leur contrepartie pour les diséquations. Mais aussi les règles d'élimination des paramètres. La règle (EP_2), par exemple, consiste (informellement) à faire le raisonnement suivant: si la diséquation $y \neq t$ n'est pas satisfaite, c'est que y égale t et qu'il est donc possible de remplacer y par t. Enfin, les règles de "mise en forme" des résultats permettent d'éliminer les inconnues auxiliaires qui ne sont plus nécessaires et de réintroduire au contraire les inconnues principales qui ont disparu.

Elimination des équations et diséquations triviales (T)

$$(T_1) \quad s = s \quad \mapsto \quad \top$$
$$(T_2) \quad s \neq s \quad \mapsto \quad \bot$$

Fusions (F)

$$(F_1) \quad s = t \wedge s = u \quad \mapsto \quad s = t \wedge t = u$$
$$(F_2) \quad s \neq t \vee s \neq u \quad \mapsto \quad s \neq t \vee t \neq u$$
$$(F_3) \quad s = t \wedge s \neq u \quad \mapsto \quad s = t \wedge t \neq u$$
$$(F_4) \quad s = t \vee s \neq u \quad \mapsto \quad u = t \vee s \neq u$$

Remplacements (R)

$$(R_1) \quad z = t \wedge P \quad \mapsto \quad z = t \wedge P\{z \to t\}$$
$$(R_2) \quad z \neq t \vee P \quad \mapsto \quad z \neq t \vee P\{z \to t\}$$

Elimination des paramètres (Ebf P)

$$(EP_1) \quad \forall \vec{y}, y : P \quad \mapsto \quad \forall \vec{y} : P \qquad \text{Si } y \notin Var(P)$$
$$(EP_2) \quad \forall \vec{y} : P \wedge (y \neq t \vee d) \quad \mapsto \quad \forall \vec{y} : P \wedge d\{y \to t\} \qquad \text{Si } d \text{ est une disjonction d'équations et de diséquations, } y \in \vec{y} \text{ et } y \notin Var(t)$$

Règles de "mise en forme" des résultats (MF)

$$(MF_1) \quad \exists w, \mathcal{P} \quad \mapsto \quad \mathcal{P} \qquad \text{Si } w \notin Var(\mathcal{P})$$
$$(MF_2) \quad \exists \vec{w}, w : w = t \wedge P \quad \mapsto \quad \exists \vec{w} : P$$

Si $w \notin Var(P, t)$ et t ne contient pas de paramètre.

$$(MF_3) \quad \exists \vec{w}, \forall \vec{y} : P \quad \mapsto \quad \exists \vec{w}, w, \forall \vec{y} : P \wedge x = w$$

Si $x \in \mathcal{I}$, $x \notin Var(P)$, w est de même sorte que x et $w \notin Var(\vec{w}, \vec{y}, P)$.

Figure 3.1: Règles fortement adéquates pour toute F-algèbre $\mathcal{A}$

Les règles de la figure 3.2 possèdent des propriétés de correction et d'adéquation qui dépendent de la F-algèbre $\mathcal{A}$ sur laquelle on cherche les solutions. Là encore, certaines règles sont très classiques comme les tests d'occurrence, la décomposition et les incompatibilités ("clash" en anglais). D'autres le sont moins, comme les règles qui permettent d'éliminer les paramètres des équations. La règle (EP_4) est par contre très simple; elle consiste essentiellement à remplacer un paramètre par toutes ses valeurs possibles.

Décompositions (D)

$$(D_1)\quad f(t_1,\ldots,t_n) = f(u_1,\ldots u_n) \;\mapsto\; t_1 = u_1 \wedge \ldots \wedge t_n = u_n$$
$$(D_2)\quad f(t_1,\ldots,t_n) \neq f(u_1,\ldots u_n) \;\mapsto\; t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n$$

Incompatibilités (I)

$$(I_1)\quad f(t_1,\ldots,t_n) = g(u_1,\ldots,u_m) \;\mapsto\; \bot \quad \text{Si } f \neq g$$
$$(I_2)\quad f(t_1,\ldots,t_n) \neq g(u_1,\ldots,u_m) \;\mapsto\; \top \quad \text{Si } f \neq g$$

Tests d'occurrence (O)

$$(O_1)\quad z = t \;\mapsto\; \bot \quad \text{Si } z \in Var(t)$$
$$(O_2)\quad z \neq t \;\mapsto\; \top \quad \text{Si } z \in Var(t)$$

Elimination des paramètres (suite) (Ebf P')

$$(EP_3)\quad \forall \vec{y}:\ P \wedge (z_1 = u_1 \vee \ldots \vee z_n = u_n \vee R) \;\mapsto\; \forall \vec{y}:\ P \wedge R$$

Si

1. Pour tout i, z_i est une variable et est distincte de u_i
2. Pour tout i, $z_i = u_i$ contient au moins une occurrence de paramètre
3. Pour tout i et tout paramètre $y \in Var(z_i, u_i)$, y est infinitaire
4. R ne contient pas d'occurrence de paramètre

$$(EP_4)\quad \forall \vec{y}:\ P \wedge Q \;\mapsto\; \forall \vec{y}:\ P \wedge Q\{y \to t_1\} \wedge \ldots \wedge Q\{y \to t_n\}$$

Si y est un paramètre de sorte $\underline{s}$ dont le support dans $\mathcal{A}$ est $\{t_1,\ldots,t_n\}$ et $y \in Var(Q)$.

Règles de "mise en forme" des résultats (suite) (**MF'**)

$$(MF_4)\quad \exists \vec{w}:\ (d_1 \vee z_1 \neq u_1) \wedge \ldots \wedge (d_n \vee z_n \neq u_n) \wedge P \;\mapsto\; \exists \vec{w}:\ P$$

Si chaque d_i est une disjonction d'équations et de diséquations, chaque z_i est une inconnue, chaque diséquation $z_i \neq u_i$ est une diséquation non triviale, les termes u_i ne contiennent pas de paramètre et il existe une variable $w \in \vec{w} \cap Var(z_1, u_1) \cap \ldots \cap Var(z_n, u_n)$ qui n'apparait pas dans P et qui est infinitaire.

Figure 3.2: Règles fortement adéquates dans certaines F-algèbres

Explosion (E)

$$(Ex_1) \quad \forall \vec{y}: P \quad \mapsto \quad \exists w_1, \ldots, w_p, \forall \vec{y}: P \wedge z = f(w_1, \ldots, w_p)$$

Si $\{w_1, \ldots, w_p\} \cap (Var(P) \cup \vec{y} \cup \mathcal{I}) = \emptyset$, $f \in F$, $f : \underline{s}_1 \times \ldots \underline{s}_n \rightarrow \underline{s}$, pour tout i, $sort(w_i) = \underline{s}_i$ et z est membre d'une équation ou d'une diséquation de P contenant une occurrence d'un paramètre.

$$(Ex_2) \quad \mathcal{P}[x \neq u] \quad \mapsto \quad \mathcal{P}[x \neq u] \wedge x = t$$

Si x est de support fini D dans $\mathcal{A} \subseteq T(F, X)$ et $t \in D$.

Choix non déterministe (NC)

$$(Nc) \quad \forall \vec{y}: P \wedge (P_1 \vee P_2) \quad \mapsto \quad \forall \vec{y}: P \wedge P_1 \qquad \text{Si } Var(P_1) \cap \vec{y} = \emptyset \text{ ou } Var(P_2) \cap \vec{y} = \emptyset$$

Figure 3.3: Règles globalement conservatives

Le dernier groupe de règles est constitué de celles qui ne sont pas conservatives mais seulement globalement conservatives. La première (l'explosion) a déjà été évoquée plus haut. Il s'agit essentiellement de faire la supposition que la solution sur la variable z est un terme dont la racine est étiquetée par f.

3.1.3 Résulats de correction et d'adéquation des règles

Notons tout d'abord que le lemme 2.5 s'applique également aux problèmes équationnels. D'autre part, la négation n'est plus présente dans les problèmes équationnels. Pour prouver la correction (resp. la conservativité, resp. l'adéquation) de $l \mapsto r$ il suffit de prouver que $\mathcal{S}(\mathcal{A}, r\sigma, \mathcal{I}) \subseteq \mathcal{S}(\mathcal{A}, l\sigma, \mathcal{I})$ (resp. $\supseteq$, resp. $=$). De même, si $l \mapsto r$ est fortement adéquate, et que les formes prénexes l' de $\neg l$ et r' de $\neg r$ sont des problèmes équationnels, alors $l' \mapsto r'$ est fortement adéquate (toujours d'après le lemme 2.5). Ainsi, la forte adéquation des règles $(T_2), (R_2), (F_2), (F_4), (D_2), (I_2), (O_2)$ est une conséquence de la forte adéquation des règles $(T_1), (R_1), (F_1), (F_3), (D_1), (I_1), (O_1)$ respectivement.

Ces propriétés seront utilisées sans mention explicite dans les preuves qui suivent.

Proposition 3.4 *Les règles de la figure 3.1 sont fortement adéquates par rapport à tous $\mathcal{A}, \mathcal{I}$.*

Preuve

Pour certaines règles, ce résultat est une conséquence directe des définitions:

- La forte adéquation des règles (T_1) et (T_2) est une conséquence de la définition 2.7

- La forte adéquation des règles de fusion est une conséquence de la transitivité de $=$ et de la remarque ci-dessus
- La forte adéquation des règles de remplacement est aussi une conséquence de la définition 2.7
- La forte adéquation de (EP_1), (MF_1) et (MF_3) est une conséquence de la définition 2.7

Il nous reste ainsi seulement deux règles à étudier : (EP_2) et (MF_2).

Correction de la règle (EP_2)

A cause des propriétés de monotonie (cf lemme 2.5), nous n'avons à prouver que la correction de la règle: $\forall \vec{y} : y \neq t \vee d \mapsto \forall \vec{y} : d\{y \rightarrow t\}$ lorsque $y \in \vec{y}$. Soit donc σ une $\mathcal{A}$-solution de $\forall \vec{y} : d\{y \rightarrow t\}$. Nous allons prouver que σ est aussi une $\mathcal{A}$-solution de $\forall \vec{y} : d \vee y \neq t$.

Soit ψ une $\mathcal{A}$-substitution quelconque dont le domaine est $\vec{y}$. Deux cas se présentent:

- *Premier cas:* $y\psi =_{\mathcal{A}} t\psi$. Alors, $\psi\sigma \equiv (\psi \circ \{y \rightarrow t\})\sigma$.
  D'autre part, $\psi\sigma$ valide $d\{y \rightarrow t\}$ par hypothèse. Par conséquent $(\psi \circ \{y \rightarrow t\})\sigma$ valide d.
  On déduit de ces deux remarques que $\psi\sigma$ valide d et donc $d \vee y \neq t$.
- *Deuxième cas:* $y\psi \neq_{\mathcal{A}} t\psi$.
  Dans ce cas, $\psi\sigma$ valide $y \neq t$ et donc valide $y \neq t \vee d$.

La règle (EP_2) est conservative

Soit σ une $\mathcal{A}$-solution de $\forall \vec{y} : y \neq t \vee d$. Nous avons à prouver que σ est ausi solution de $\forall \vec{y} : d\{y \rightarrow t\}$.
Soit ψ une $\mathcal{A}$-substitution quelconque dont le domaine est $\vec{y}$. $\{y \rightarrow t\sigma\}\psi$ est une $\mathcal{A}$-substitution de même domaine. Comme σ est une $\mathcal{A}$-solution de $\forall \vec{y} : d \vee y \neq t$, $\sigma\{y \rightarrow t\sigma\}\psi$ valide $y \neq t \vee d$. De plus, comme $\sigma\{y \rightarrow t\sigma\}\psi$ ne peut valider $y \neq t$, cette substitution valide d. Enfin, $\sigma\{y \rightarrow t\sigma\}\psi \equiv (\sigma \circ \{y \rightarrow t\})\psi$ et par suite $\sigma\psi$ valide d.

Forte adéquation de (MF_2)

Il suffit de montrer que la règle est correcte car l'inclusion $\mathcal{S}(\mathcal{A}, \exists \vec{w}, w : w = t \wedge P, \mathcal{I}) \subseteq \mathcal{S}(\mathcal{A}, \exists \vec{w} : P, \mathcal{I})$ est triviale. De même que précédemment, P ne joue aucun rôle et peut donc être supposé équivalent à $\top$. Il s'agit donc de prouver que toute $\mathcal{A}$-substitution de domaine $\mathcal{I}$ est une solution de $\exists \vec{w}, w : w = t$. Si σ est une $\mathcal{A}$-substitution et ψ une $\mathcal{A}$-substitution de domaine $\vec{w}$, $\sigma\{w \rightarrow t\sigma\psi\}\psi$ valide $w = t$ puisque $w \notin Var(t)$. Par conséquent, il existe une substitution ψ' ($\psi' \equiv \{w \rightarrow t\sigma\psi\}\psi$) telle que $\sigma\psi$ valide $w = t$. C'est ce que nous voulions montrer.

□

Proposition 3.5 *Les règles* **I**, **D**, **O**, (EP_3) *et* (MF_4) *sont fortement adéquates lorsque $\mathcal{A}$ est une sous-algèbre de $T(F, X)$. La règle (EP_4) est fortement adéquate lorsque $\mathcal{A}$ est ou bien une sous-algèbre de $T(F, X)$, ou bien un quotient d'une sous-algèbre de $T(F, X)$.*

Notons que certaines règles (comme les décompositions et incompatibilités) sont fortement adéquates dans une classe plus large de modèles que nous ne rappelons pas ici (cf [BHS87]).

Pour prouver la correction de la règle (EP_3), nous avons besoin de deux lemmes techniques. Le premier concerne l'expression des solutions dans $T(F,X)$ d'une équation et peut être vu comme un cas particulier des résultats de [LMM86]. Le deuxième lemme est plus fondamental : nous le réutiliserons par la suite car il permet d'assurer l'existence d'une solution à certains systèmes de diséquations. Des résultats semblables à ce lemme sont d'ailleurs donnés dans [Col82,LMM86,Mah88a]. Ce résultat se généralise à certaines théories équationnelles (cf chapitre 7).

Commençons par énoncer et prouver ces lemmes.

Lemme 3.6 *Une équation $s = t$ dont la seule variable est x possède au plus une solution dans $T(F)$.*

Preuve

Nous effectuerons la preuve par récurrence sur la profondeur minimale de s et t.

- Si s ou t sont de profondeur 0, l'un des deux termes (s par exemple) est une variable ou une constante. Si s est une constante, alors - ou bien $t \equiv x$ et l'équation $s = t$ a pour seule solution $\{x \rightarrow s\}$ -ou bien $t \not\equiv x$ et $s = t$ n'a pas de solution car $t\sigma/\epsilon \not\equiv s$ (Rappelons que x est une variable de t et que c'est sa seule variable).

- Si s et t sont tous deux de profondeur au moins 1, $s \equiv g(s_1, \ldots s_m)$ et $t \equiv f(t_1, \ldots, t_m)$. Si $f \neq g$, l'équation n'a alors pas de solution. Sinon, il existe un indice i tel que $x \in Var(s_i, t_i)$. Une solution dans $T(F)$ de $s = t$ est aussi une solution de $s_i = t_i$. Par hypothèse de récurrence une telle équation possède au plus une solution dans $T(F)$. Par conséquent, il en est de même de $s = t$. □

Lemme 3.7 *Soit $\mathcal{P}$ une conjonction de diséquations non triviales. Soit A un sous-ensemble non vide de $T(F,X)$ tel que $Var(\mathcal{P}) \cap A = \emptyset$. On suppose de plus que toute variable de $\mathcal{P}$ a une sorte dont le support dans A est infini. Alors $\mathcal{P}$ a au moins une solution dans A.*

Preuve du lemme 3.7

Pour plus de simplicité, nous allons supposer que A est un sous-ensemble de $T(F)$, ce que l'on peut toujours supposer en considérant les éléments de $A \cap X$ comme des constantes. (La nouvelle signature peut éventuellement contenir une infinité de symboles fonctionnels, mais cela n'a pas ici d'importance).

Nous allons effectuer la preuve par récurrence sur le nombre de variables distinctes de $\mathcal{P}$.

- Si $\mathcal{P}$ ne contient pas de variable, alors les diséquations sont toutes trivialement valides dans $\mathcal{A} \subseteq T(F,X)$. Toute substitution dans A est alors solution.

- Supposons maintenant que la propriété est vraie pour $|Var(\mathcal{P})| \leq m - 1$ $(m \geq 1)$ et considérons un problème $\mathcal{P}$ vérifiant les hypothèse du lemme et tel que $|Var(\mathcal{P})| =$

m. Soit $x \in Var(\mathcal{P})$. Soit $s \neq t$ une diséquation de $\mathcal{P}$ et (S, F') la signature obtenue en considérant les variables de $\mathcal{P}$ autres que x comme des constantes. D'après le lemme 3.6, $s = t$ a au plus une solution dans $T(F')$.

Soit $\mathcal{S}$ l'ensemble (fini) des solutions dans $T(F')$ de telles équations. Comme il y a dans A une infinité de termes de même sorte que x, il existe dans A un terme t_x de même sorte que x et n'appartenant pas à $\mathcal{S}$. Le problème $\mathcal{P}'$ obtenu en substituant t_x à x dans $\mathcal{P}$ est alors une conjonction de diséquations non triviales, par construction de t_x. Par hypothèse de récurrence, $\mathcal{P}'$ a au moins une solution σ dans A. $\sigma \circ \{x \rightarrow t_x\}$ est alors une solution dans A de $\mathcal{P}$. □

Preuve de la proposition 3.5

La forte adéquation des règles **D**, **O** et **I** dans les sous-algèbres de $T(F, X)$ est triviale. En fait, c'est une conséquence du fait que ces algèbres sont *localement libres* [Mal71].

La règle (MF_4) est d'autre part trivialement conservative, comme la règle (EP_3) est trivialement correcte.

Enfin, nous supposerons que $\mathcal{A} \cap X = \emptyset$. (Ce que l'on peut toujours faire au prix d'un éventuel changement de signature).

Correction de (MF_4)

Par propriété de monotonie, on peut supposer que $\vec{w} = \{w\}$. De plus, comme $w \notin Var(P)$, le membre gauche peut aussi s'écrire

$$(\exists w : (d_1 \vee z_1 \neq u_1) \wedge \ldots \wedge (d_n \neq u_n)) \wedge P$$

et le membre droit est équivalent à P. A nouveau par propriété de monotonie, il nous suffit de prouver que la règle

$$\exists w : (d_1 \vee z_1 \neq u_1) \wedge \ldots \wedge (d_n \neq u_n) \mapsto \top$$

est correcte. Le membre gauche peut encore s'écrire $\exists w : R \vee (z_1 \neq u_1 \wedge \ldots \wedge z_n \neq u_n)$ pour un certain R. Il suffit donc de prouver que $\exists w : z_1 \neq u_1 \wedge \ldots \wedge z_n \neq u_n \mapsto \top$ est correcte (à nouveau par monotonie, et puisque les quantificateurs existentiels "traversent" les disjonctions).

Soit σ une $\mathcal{A}$-substitution quelconque de domaine $Var(z_1, u_1, \ldots, z_n, u_n) - \{w\}$. Soit $\mathcal{P}' \equiv z_1\sigma \neq u_1\sigma \wedge \ldots \wedge z_n\sigma \neq u_n\sigma$. $\mathcal{P}'$ est une conjonction de diséquations non triviales puisque, pour tout i, -ou bien $z_i \equiv w$ et $z_i\sigma \equiv z_i \equiv w$ est distinct de $u_i\sigma$ par hypothèse -ou bien $z_i \not\equiv w$. Dans ce dernier cas, $w \in Var(u_i\sigma)$ et $w \notin Var(z_i\sigma)$. Par application du lemme 3.7 il existe donc une solution dans $\mathcal{A}$ à $\mathcal{P}'$. c'est-à-dire une substitution ψ de domaine $\{w\}$ telle que, pour tout i, $z_i\sigma\psi =_{\mathcal{A}} u_i\sigma\psi$. Il en résulte que toute $\mathcal{A}$-substitution est une solution de $z_1 \neq u_1 \wedge \ldots \wedge z_n \neq u_n$ (c'est ce que l'on voulait prouver).

Forte adéquation de (EP_3)

Tout d'abord, P n'est pas significatif dans cette propriété et peut donc être supposé égal à $\top$. En prenant la négation des deux membres, on est ramené à prouver la

correction de la règle $\exists \vec{y} : z_1 \neq u_1 \wedge \ldots \wedge z_n \neq u_n \mapsto \top$. Comme ci-dessus, si σ est une $\mathcal{A}$-substitution quelconque, il suffit d'appliquer σ au problème; on peut alors utiliser le lemme 3.7 assurant l'exitence d'une substitution de domaine $\vec{y}$ qui valide la formule. [1]

Forte adéquation de (EP_4)

Elle est triviale. Remarquons seulement que -d'une part cette règle est aussi adéquate dans tout quotient de $T(F,X)$ -d'autre part, un léger problème technique se pose lorsque $t_1, \ldots, t_n$ contiennent des variables: les nouvelles variables ainsi introduites sont des variables libres et pourtant non contenues dans $\mathcal{I}$. Il faut en effet considérer ici ces nouvelles variables comme des constantes vis-à-vis de la résolution des problèmes. Nous n'entrerons pas ici dans le détail d'une formalisation possible qui apporterait plus de complications que d'éclaircissements.

□

Proposition 3.8 *La règle (Ex_1) est $\mathcal{A}$-adéquate lorsque $\mathcal{A}$ est soit un sous-ensemble de $T(F)$, soit $RT(F)$, soit un quotient de $RT(F)$. La règle (Ex_2) est $\mathcal{A}$-adéquate lorsque $\mathcal{A} \subseteq T(F,X)$. (Nc) est $\mathcal{A}$-adéquate pour tout $\mathcal{A}$.*

La règle (Ex_1) correspond au "Domain Closure Axiom" [Rei78,Mah88a]. Elle exprime en effet que tous les éléments du domaine sont construits avec l'un des symboles de fonction de la signature.[2] Il est donc naturel qu'elle ne soit globalement conservative que lorsque tous les éléments de l'algèbre $\mathcal{A}$ peuvent être "atteints" en utilisant les symboles de F seulement.

Preuve

En ce qui concerne la règle (Ex_1)

La correction est immédiate. De plus, comme dans les cas précédents, il suffit de prouver la globale conservation de

$$\top \mapsto \exists w_1, \ldots, w_p, \forall \vec{y} : z = f(w_1, \ldots, w_p)$$

puisque les variables $w_1, \ldots, w_p$ n'ont pas d'occurrence dans P. (Rappelons à nouveau que les solutions de $\vec{y} : P \wedge Q$ sont les substitutions qui sont solution à la fois de $\forall \vec{y} : P$ et de $\forall \vec{y} : Q$). On peut de plus supposer $\vec{y} = \emptyset$ sans perdre de généralité puisque $z = f(w_1, \ldots, w_p)$ ne contient pas de paramètre. (C'est une application de la règle (EP_1)).

Il reste alors à remarquer que la globale conservation est une conséquence des hypothèses effectuées sur l'algèbre $\mathcal{A}$. En effet, si σ est une $\mathcal{A}$-substitution quelconque

[1] On n'est pas vraiment ramené à la correction de (MF_4) car $z_1, u_1, \ldots, z_n, u_n$ peuvent contenir respectivement des occurrences de différents paramètres. Mais une application répétée de la règle (MF_4) que nous avons prouvée correcte permet aussi de prouver la forte adéquation de (ER)

[2] On utilise ici (et seulement ici) que F est fini.

de domaine contenant z, $z\sigma$ peut s'écrire $f(s_1, \ldots, s_p)$ pour un certain f [3]. On considère alors l'instance de la règle (Ex_1) qui correspond à ce f là. (Rappelons que nons n'avons à prouver que la *globale* conservation). Si $\theta = \{w_1 \to s_1; \ldots; w_p \to s_p\}$, $\sigma\theta$ valide $z = f(w_1, \ldots, w_p)$, ce qui prouve bien que σ est une solution du membre droit.

En ce qui concerne la règle (Ex_2). L'adéquation est immédiate.

En ce qui concerne la règle (Nc) La correction est triviale. Nous ne nous intéresserons qu'à la globale conservation. P_1 et P_2 jouent des rôles symétriques. En effet, $\vee$ étant commutatif, $\forall \vec{y} : P \wedge (P_1 \vee P_2) \mapsto \forall \vec{y} : P_2$ n'est qu'une instance de (Nc). Sans perdre de généralité on peut donc supposer que $\vec{y} \cap Var(P_1) = \emptyset$. De même, P ne jouant aucun rôle peut être supposé égal à $\top$. Il suffit ainsi de prouver que toute solution de $\forall \vec{y} : P_1 \vee P_2$ est une solution de P_1 ou une solution de $\forall \vec{y} : P_2$ (qui correspondent à deux instances d'application de la règle). Mais (comme il a été rappelé dans le chapitre 2) $\forall \vec{y}, \vec{y'} : \phi_1(\vec{y} \to \vec{y'}) \vee \phi_2$ et $(\forall \vec{y} : \phi_1) \vee (\forall \vec{y} : \phi_2)$ sont deux formules équivalentes. On obtient alors la relation voulue en remarquant qu'ici $\vec{y'} = \emptyset$.

□

3.2 Transformation de représentants de problèmes

L'algèbre des problèmes équationnels telle qu'elle est décrite dans la figure 2.1 est une algèbre quotient. Bien sûr, les règles de transformation de la section précédente sont compatibles avec cette structure quotient. Si bien que l'on peut appliquer les règles à n'importe quel représentant d'une classe d'équivalence de problèmes. Néanmoins, la décision de l'applicabilité d'une règle à un problème entrainerait une opération de filtrage modulo les équations de la figure 2.1. D'un point de vue pratique une telle opération serait extrêmement coûteuse (si elle est décidable ![4]). De plus, il ne serait pas aisé de donner des preuves de terminaison, les ordres uitilisés devant alors être compatibles avec la structure quotient.

Il apparait ainsi qu'il peut être utile de choisir des *représentants* canoniques des problèmes équationnels, l'applicabilité d'une règle se ramenant alors à un problème de filtrage modulo des équations de commutativité-associativité.

Nous choisirons donc dans ce chapitre (et le suivant) d'effectuer une "normalisation" des problèmes équationnels avant l'application d'une règle de transformation; les problèmes auxquels sont appliqués les règles seront supposés en *forme normale conjonctive*. Ces formes normales sont décrites dans de nombreux livres de logique (cf par exemple [Gal86]). Elles sont obtenues en orientant les axiomes de la figure 2.1 (autres que l'associativité et la commutativité). L'orientation des axiomes d'idempotence, élément absorbant, élément neutre est évidente. Les axiomes de distributivité sont orientés de façon à faire apparaitre

[3] Ici, f désigne aussi bien la fonction $f_{\mathcal{A}}$ que le symbole $f \in F$.

[4] Noter en effet que les classes d'équivalence de problèmes sont des ensembles infinis

(Dis1)	$P_1 \vee (P_2 \wedge P_3)$	$\rightarrow$	$(P_1 \wedge P_2) \vee (P_1 \wedge P_3)$
(Dis2)	$P_1 \wedge (P_1 \vee P_2)$	$\rightarrow$	P_1
(Idem1)	$P \wedge P$	$\rightarrow$	P
(Idem2)	$P \vee P$	$\rightarrow$	P
(Neu1)	$P \vee \bot$	$\rightarrow$	P
(Neu2)	$P \wedge \top$	$\rightarrow$	P
(Abs1)	$P \vee \top$	$\rightarrow$	$\top$
(Abs2)	$P \wedge \bot$	$\rightarrow$	$\bot$

Figure 3.4: Mise en forme normale conjonctive des problèmes équationnels

le symbole $\wedge$ à une position de taille plus petite. Ces règles sont exposées dans la figure 3.4. Un problème en forme normale conjonctive sera ainsi de la forme $\exists \vec{w}, \forall \vec{y} : P$ où P est une *matrice* , c'est-à-dire une conjonction de disjonctions d'équations et de diséquations.

Malheureusement, la normalisation des problèmes avant chaque application de règle peut empêcher l'application d'une règle. Par exemple, considérons le problème $(z = t \wedge z = u) \vee (z = t \wedge d)$:

$$(z = t \wedge z = u) \vee (z = t \wedge d) \rightarrow_{\text{normalisation}} z = t \wedge (z = u \vee d)$$

MAIS

$$\begin{array}{lll} (z = t \wedge z = u) \vee (z = t \wedge d) & \mapsto_{\text{Fusion}} & (z = t \wedge t = u) \vee (z = t \wedge d) \\ & \rightarrow_{\text{normalisation}} & z = t \wedge (t = u \vee d) \end{array}$$

nous voyons ici que la normalisation du problème peut empêcher l'application d'une règle de fusion. Ce problème s'apparente à celui de la confluence des systèmes de réécriture: l'orientation d'une règle (ici la mise en forme normale conjonctive) peut restreindre l'ensemble des formules prouvables. Classiquement, l'algorithme de Knuth-Bendix [DJ88, BDH86] permet justement de "compléter" le système pour "récupérer" la confluence. Nous allons procéder de façon analogue en rajoutant certaines règles (comme par exemple la règle $z = t \wedge (z = u \vee d) \mapsto z = t \wedge (t = u \vee d)$ qui permet dans l'exemple ci-dessus de récupérer la transformation rendue impossible par la normalisation). Nous n'essaierons pas de donner un système confluent (i.e. nos n'essaierons pas d'être exhaustifs dans la complétion) mais seulement de donner un système qui nous suffira pour les preuves de complétude. Ces "nouvelles règles" sont réunies dans la figure 3.5.

Il faut aussi ajouter aux règles de la figure 3.5 celles que l'on obtient en remplaçant dans une règle une ou plusieures variables de sorte *système* par $\top$ ou $\bot$ puis en normalisant les problèmes obtenus. Nous ne reproduisons pas ici les 14 règles obtenues de cette façon. Nous considérerons plutôt que les règles de transformation dénotent l'ensemble des règles de transformation obtenues ainsi. Il faudra le garder à l'esprit dans les preuves de terminaison.

$$(F_1') \quad z = t \wedge (z = u \vee d) \mapsto z = t \wedge (t = u \vee d)$$
$$(F_3') \quad z = t \wedge (z \neq u \vee d) \mapsto z = t \wedge (t \neq u \vee d)$$

Figure 3.5: Règles de transformation issues de l'"interaction" avec la mise en forme normale conjonctive

$$(D_3) \quad f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n) \vee d \mapsto (t_1 = u_1 \vee d) \wedge \ldots \wedge (t_n = u_n \vee d)$$

Figure 3.6: Règles obtenues par enchainement d'une transformation et d'une normalisation

On peut aussi remarquer que certaines règles transforment un problème en forme normale conjonctive en un problème qui n'est pas en forme normale conjonctive. Il faudra donc aussi tenir compte dans des preuves de terminaison de la normalisations entre chaque transformation. En fait, cela ne posera de problème que pour la règle de décomposition D_1, lorsqu'elle est appliquée à une équation apparaissant dans une disjonction. Dans tous les autres cas, les seules règles de normalisation éventuellement applicables après transformation d'un problème en forme normale conjonctive sont des règles qui trivialement "simplifient le problème" comme les règles d'élément absorbant ou d'élément neutre. C'est pourquoi nous ajoutons (seulement) la règle D_3 (cf figure 3.6) en combinant D_1 et la normalisation.

3.3 Contrôle

Dans cette section et les suivantes, nous ne considérerons que des formes normales conjonctives. Cette propriété sera donc sous-entendue lorsque nous parlerons de problèmes équationnels.

Les règles des sections précédentes, en général, ne terminent pas. Nous nous intéressons donc à des restrictions dans l'emploi des règles (contrôle) qui permettent d'obtenir des problèmes "plus simples" que nous appelerons formes résolues. Mais qu'entend on exactement par "plus simple" ? C'est ce que nous allons préciser dans cette section. Nous introduirons aussi la notion de complétude par rapport à un ensemble de formes résolues. Enfin, nous ferons quelques rappels sur les ordres que nous utiliserons dans les preuves de terminaison.

3.3.1 Formes résolues

Comme nous l'avons vu au début de ce chapitre, un ensemble de règles de transformation définit une relation de réduction ($\mapsto$) dans l'algèbre des problèmes équationnels. Un *contrôle* permet de définir une relation est moins fine que $\mapsto$, relation qui peut aussi être

vue comme une relation de réduction associée à un système de réécriture. (Nous noterons encore $\mapsto$ cette relation).

Un tel contrôle sera défini d'une part par une restriction d'emploi des règles de transformation, d'autre part par un ordre sur les règles qui détermine quelles règles appliquer en priorité lorsqu'il se présente plusieurs possibilités. [5]

Remarquons que, si la relation de réduction associée à un ensemble de règles de transformation est fortement adéquate (resp. correcte), alors la relation de réduction obtenue en ajoutant un contrôle est elle-même fortement adéquate (resp. correcte). De plus, tous les contrôles que nous utiliserons préserveront l'adéquation des règles. [6]

Remarquons enfin qu'il n'y a qu'un nombre fini de façons d'appliquer une même règle à un problème équationnel en forme normale conjonctive, car un tel problème n'est égal qu'à un nombre fini de formes normales conjonctives.

Pour être intéressant, un système de règles de transformation doit permettre d'aboutir, à partir d'un problème équationnel, à un ensemble de *formes résolues* ayant de "bonnes propriétés". Commençons par préciser ce que peuvent être de "bonnes propriétés".

Définition 3.9 *Soient $\mathcal{A}$ une F-algèbre et $\mathcal{I}$ un ensemble fini de variables. Deux ensembles de problèmes équationnels (éventuellement infinis) $\{\mathcal{P}_i | i \in I_1\}$ et $\{\mathcal{P}_i | i \in I_2\}$ sont dits* équivalents [7] *par rapport à $\mathcal{A}$ et $\mathcal{I}$ si*

- *Pour tout $i \in I_1 \cup I_2$, les variables libres de $\mathcal{P}_i$ sont contenues dans $\mathcal{I}$*
- $\bigcup_{i \in I_1} \mathcal{S}(\mathcal{A}, \mathcal{P}_i, \mathcal{I}) = \bigcup_{i \in I_2} \mathcal{S}(\mathcal{A}, \mathcal{P}_i, \mathcal{I})$.

Cette définition n'est qu'une généralisation de la définition de l'équivalence de deux problèmes donnée dans le chapitre 2.

Une première "bonne propriété" de l'ensemble des formes résolues d'un problème $\mathcal{P}$ est qu'il soit équivalent à $\mathcal{P}$. Cela correspond à l'adéquation des règles.

Définition 3.10 *Un problème équationnel est dit* sans paramètre *s'il ne contient pas le quantificateur $\forall$*

Une deuxième "bonne propriété" peut être ainsi l'absence de quantificateur universel. On pourra donc considérer qu'un problème est en forme résolue s'il est sans paramètre. Par exemple, nous utiliserons ce genre de forme résolue pour décider de la validité d'une formule équationnelle dans $\mathcal{A}$.[8]

Définition 3.11 *Un problème équationnel $\mathcal{P}$ est dit* soluble *(par rapport à $\mathcal{A}$ et $\mathcal{I}$) si $\mathcal{S}(\mathcal{A}, \mathcal{P}, \mathcal{I}) \neq \emptyset$,*

[5] En fait, cette dernière façon d'exprimer le contrôle n'est qu'un cas particulier de la première: on restreint l'emploi d'une règle aux cas où aucune règle plus petite n'est applicable.

[6] Nous faisons ici allusion aux règles qui sont adéquates sans être fortement adéquates.

[7] Cette notion d'équivalence n'est qu'une extension aux ensembles de problèmes de la relation $\approx_{\mathcal{A},\mathcal{I}}$ définie dans le chapitre précédent.

[8] Transformer un problème en problèmes sans paramètres correspond à l'élimination du quantificateur "le plus interne" dans une formule équationnelle.

Une autre bonne propriété d'une forme résolue est qu'elle soit ou bien (syntaxiquement) égale à $\bot$ ou bien soluble. Cette propriété de solubilité est assurée lorsque le problème équationnel considéré est une *définition contrainte* :

Définition 3.12 *Lorsque $\mathcal{A}$ est une sous-algèbre de $T(F,X)$, un problème équationnel $\mathcal{P}$ est dit être une* définition contrainte *(par rapport à $\mathcal{A}$ et $\mathcal{I}$) si $\mathcal{P}$ est ou bien $\top$ ou $\bot$ ou bien une conjonction d'équations et de diséquations de la forme*

$$\exists w_1, \ldots, w_k : x_1 = t_1 \wedge \ldots \wedge x_m = t_m \wedge x'_1 \neq u_1 \wedge \ldots \wedge x'_p \neq u_p$$

où

1. *$x_1, \ldots, x_m$ sont des variables et n'apparaissent qu'une fois dans $\mathcal{P}$*
2. *pour tout indice $1 \leq i \leq p$, x'_i est infinitaire et est distinct de u_i*

Proposition 3.13 *Une définition contrainte distincte de $\bot$ a au moins une solution dans $T(F)$.*

Preuve
Soit $\mathcal{P} \equiv \exists w_1, \ldots, w_k : x_1 = t_1 \wedge \ldots \wedge x_m = t_m \wedge x'_1 \neq u_1 \wedge \ldots \wedge x'_p \neq u_p$ une définition contrainte (non triviale). D'après le lemme 3.7 (la définition 3.12 assure que les hypothèses de ce lemme sont satisfaites), il existe au moins une $T(F)$-solution θ à $x'_1 \neq u_1 \wedge \ldots \wedge x'_p \neq u_p$. Soit alors $\sigma = \{x_1 \rightarrow t_1; \ldots; x_m \rightarrow t_m\}$. $\theta \circ \sigma$ restreinte aux inconnues principales du problème est alors une $T(F)$-solution de $\mathcal{P}$ (à cause de la propriété d'occurrence unique des x_i dans la définition 3.12). □

Définition 3.14 *Un problème équationnel $\mathcal{P}$ est appelé* problème d'unification *s'il ne contient ni diséquation (autre que $\bot$) ni quantificateur universel.*

Un problème d'unification est ainsi un cas particulier de problème sans paramètre. Dans un tel problème, il n'y a pas de négation. Une "bonne propriété" d'une forme résolue de $\mathcal{P}$ est d'être un problème d'unification, lorsque $\mathcal{P}$ est équivalent à un ensemble fini de problèmes d'unification. De telles propriétés sont étudiées dans [LMM86,LM87]. Elles sont utiles dans l'apprentissage par exemples et contre-exemples ([LM87]) mais aussi dans d'autres applications étudiées dans le chapitre 6.

Nous voyons donc que plusieurs définitions de la notion de forme résolue sont intéressantes. De plus (quelle que soit cette définition) il n'est pas nécessaire de calculer toutes les formes résolues d'un problème mais seulement un sous-ensemble équivalent à l'ensemble entier. Pour une définition donnée de forme résolue, on souhaite que l'ensemble de règles de transformation soit *complet* par rapport à ces formes résolues:

Définition 3.15
Un système $\mathcal{R}$ de règles de transformation est dit complet *par rapport à :*

- *Un ensemble de formes initiales $\mathcal{F}_I$ contenu dans l'ensemble des problèmes équationnels*

- *Une application $\mathcal{F}_R$ (Formes Résolues) de $\mathcal{F}_I$ dans l'ensemble des parties de $\mathcal{F}_I$*
- *Une F-algèbre $\mathcal{A}$*

si, pour tout problème $\mathcal{P} \in \mathcal{F}_I$, il existe un ensemble $\mathcal{F}_{RS}(\mathcal{P}) \subseteq \mathcal{F}_R(\mathcal{P})$ qui est équivalent (par rapport à $\mathcal{A}$ et $\mathcal{I} = VL(\mathcal{P})$) à $\mathcal{F}_R(\mathcal{P})$ et tel que

$$\forall \mathcal{P}' \in \mathcal{F}_{RS}(\mathcal{P}),\ \mathcal{P} \mapsto_{\mathcal{R}}^{*} \mathcal{P}'$$

Par abus, nous parlerons aussi d'*ensemble de formes résolues* $\mathcal{F}_R$ au lieu de l'application $\mathcal{F}_R$. Il est alors sous-entendu que l'application $\mathcal{F}_R$ associe à un problème $\mathcal{P}$ l'ensemble des problèmes $\mathcal{P}'$ en forme résolue dont les solutions sont aussi des solutions de $\mathcal{P}$.

Notre objectif sera dans la suite de donner des systèmes de règles de transformation complets correspondant à différentes notions de formes résolues. [9]

Dans le cas où $\mathcal{A}$ est une algèbre libre, nous obtiendrons toujours un système de transformation à terminaison finie. La complétude se prouve alors assez facilement, d'où la

Remarque fondamentale pour la démarche suivie
Si $\mathcal{R}$ est un ensemble de règles de transformation adéquat qui termine et que toute forme irréductible pour $\mathcal{R}$ est en forme résolue, alors $\mathcal{R}$ est complet.

Il suffit en effet de remarquer que, $\mathcal{R}$ étant adéquat, pour tout problème $\mathcal{P}$ de $\mathcal{F}_I$ et tout entier n, l'ensemble des problèmes $\mathcal{P}'$ tels qu'existent $\mathcal{P}_1, \ldots \mathcal{P}_n$ vérifiant

$$\mathcal{P} \mapsto_{\mathcal{R}} \mathcal{P}_1 \mapsto_{\mathcal{R}} \ldots \mapsto_{\mathcal{R}} \mathcal{P}_k \equiv \mathcal{P}' \text{ avec } k \leq n$$

est équivalent à $\mathcal{P}$. Comme, de plus, $\mathcal{R}$ est à terminaison finie, pour tout problème $\mathcal{P} \in \mathcal{F}_I$, il existe un entier $n(\mathcal{P})$ tel que tout problème qui s'obtient par une succession de transformations à partir de $\mathcal{P}$ s'obtient par au plus $n(\mathcal{P})$ transformations à partir de $\mathcal{P}$.

---

[9] **Remarque**
Lorsqu'une règle est adéquate sans être fortement adéquate il faut, en général, calculer *toutes* les façons possibles d'appliquer la règle pour conserver toutes les solutions du problème. Un algorithme de transformation complet manipulera ainsi des *ensembles* de problèmes équationnels (en forme normale conjonctive). La transformation de tels ensembles peut par contre tenir compte de la forte adéquation de certaines règles: si $\mathcal{E}$ est un ensemble (fini) de problèmes équationnels et $\mathcal{R}$ est un ensemble (fini) de règles de transformation, on dira que $\mathcal{E}$ *se transforme en* $\mathcal{E}'$ en une étape (ce que l'on écrit $\mathcal{E} \mapsto_{\mathcal{R}} \mathcal{E}'$) si $\mathcal{E}' = \bigcup_{\mathcal{P} \in \mathcal{E}} \mathcal{E}_{\mathcal{P},\mathcal{R}}$ où $\mathcal{E}_{\mathcal{P},\mathcal{R}}$ est défini par:

- $\mathcal{E}_{\mathcal{P},\mathcal{R}}$ est le seul problème $\{\mathcal{P}'\}$ si la seule règle applicable est une règle fortement adéquate $\mathcal{R}_0$ et que $\mathcal{P} \mapsto_{\mathcal{R}_0} \mathcal{P}'$
- $\mathcal{E}_{\mathcal{P},\mathcal{R}} = \{\exists w_1, \ldots, w_p : \mathcal{P} \wedge z = f(w_1, \ldots, w_p) \mid f \in F\}$ si la seule règle applicable à $\mathcal{P}$ est l'explosion (Ex_1) (on peut donner une définition analogue pour (Ex_2)).
- $\mathcal{E}_{\mathcal{P},\mathcal{R}} = \{\exists \vec{w}, \forall \vec{y} : P \wedge P_1 \ ; \ \exists \vec{w}, \forall \vec{y} : P \wedge P_2\}$ si $\mathcal{P} \equiv \exists \vec{w}, \forall \vec{y} : P \wedge (P_1 \vee P_2)$ et que la seule règle applicable est le choix non déterministe (Nc).
- $\mathcal{E}_{\mathcal{P},\mathcal{R}} = \mathcal{E}_{\mathcal{P},\mathcal{R}_1} \cup \ldots \cup \mathcal{E}_{\mathcal{P},\mathcal{R}_k}$ si $\mathcal{R}_1, \ldots \mathcal{R}_k$ sont les règles de $\mathcal{R}$ applicables à $\mathcal{P}$.

On peut remarquer alors que, si $\mathcal{E} \mapsto_{\mathcal{R}} \mathcal{E}'$ et si $\mathcal{R}$ est adéquat, alors $\mathcal{E}$ et $\mathcal{E}'$ sont équivalents.

Cette remarque peut être importante d'un point de vue pratique car l'utilisation non déterministe des règles conduit rapidement à une explosion combinatoire.

De même, en pratique, pour éviter au maximum d'avoir à manipuler des ensembles de problèmes, il sera préférable de préciser un ordre total sur les règles de façon à ce qu'il n'y en ait qu'une d'applicable (au plus) à un problème donné. Dans les exemples, nous utiliserons toujours un tel système.

Il en résulte alors que l'ensemble des problèmes irréductibles obtenus par transformation de $\mathcal{P}$ est équivalent à $\mathcal{P}$. Notons $\mathcal{F}_{RS}$ cet ensemble. Par propriété des formes résolues, $\mathcal{F}_R(\mathcal{P})$ est équivalent à $\mathcal{F}_{RS}(\mathcal{P})$, ce qui prouve bien la complétude de $\mathcal{R}$.

Ceci nous indique la démarche à suivre dans les cas où il y a terminaison:

1. S'assurer de l'adéquation
2. Prouver la terminaison
3. Prouver que les formes irréductibles sont des formes résolues

3.3.2 Ordres utilisés dans les preuves de terminaison

Nous utiliserons dans les preuve de terminaison des ordres bien fondés construits par extensions lexicographiques et/ou multi-ensemble. Nous rappelons brièvement ici les principales définitions (et résultats) utilisés dans la suite.

Un ordre défini sur D est *bien fondé* s'il n'y a pas de chaine infinie strictement décroissante dans D.

Si $D_1, \ldots, D_n$ sont des ensembles, chaque D_i étant muni d'une relation d'ordre $\geq_i$, la *composée lexicographique* $\geq_{lex}$ des ordres $\geq_i$ est la relation d'ordre définie sur le produit cartésien $D_1 \times \ldots \times D_n$ par:

$$(a_1, \ldots, a_n) >_{lex} (b_1, \ldots, b_n) \Leftrightarrow \exists i \in \{1, \ldots n\}, \forall j < i, a_j = b_j \text{ et } a_i >_i b_i$$

$\geq_{lex}$ possède les propriétés suivantes:

- Si chacune des relations d'ordre $\geq_i$ est totale, alors $\geq_{lex}$ est totale.
- Si chacun des ordres $\geq_i$ est bien fondée sur D_i, alors $\geq_{lex}$ est bien fondé sur D.

Si D est un ensemble muni d'une relation d'ordre $\geq$, un *multi-ensemble* fini[10] d'éléments de D est une application de D dans l'ensemble des entiers naturels qui vaut 0, sauf pour un nombre fini d'éléments de D.

Le multi-ensemble M qui associe à $a_1, \ldots, a_n \in A$ respectivement $m_1, \ldots m_n$ et 0 à tout autre élément x de D est habituellement noté entre accolades, en répétant chaque a_i m_i fois, Par exemple, si $M(a) = 2$, $M(b) = 3$ et $M(x) = 0$ pour tout autre élément de D, on note $M = \{a, a, b, b, b\}$ (l'ordre dans lequel sont écrits les éléments n'intervient pas).

L'intersection et la réunion des multi-ensembles sont définies par :

$$\begin{aligned} (M_1 \cup M_2)(x) &= max(M_1(x), M_2(x)) \\ (M_1 \cap M_2)(x) &= min(M_1(x), M_2(x)) \end{aligned}$$

[10]Tous les multi-ensembles que nous considéreons seront finis. Nous ne le rappelerons généralement pas.

La somme et la différence de multi-ensembles sont définis par:

$$\begin{aligned}(M_1 + M_2)(x) &= M_1(x) + M_2(x)\\ (M_1 - M_2)(x) &= \max(0, M_1(x) - M_2(x))\end{aligned}$$

Enfin, on définit l'extension multi-ensemble $\geq_m$ de $\geq$ par:

$$X = \{x_1, \ldots, x_n\} \geq_m \{y_1, \ldots, y_m\} = Y$$

ssi l'une des conditions suivantes est satisfaite:

1. $X = Y$
2. $\exists i \in \{1, \ldots, n\}, \exists j \in \{1, \ldots, m\}, x_i = y_j$ et $X - \{x_i\} \geq_m Y - \{y_j\}$
3. $\exists Z \subseteq Y, \exists x \in X, \forall y \in Z,\ x > y$ et $X - \{x\} \geq_m Y - Z$

Les résultats suivants sont bien connus (cf [DM79] par exemple) ;

- Si $\geq$ est un ordre total sur D alors $\geq_m$ est un ordre total sur l'ensemble des multi-ensembles sur D.
- $\geq$ est bien fondé ssi $\geq_m$ est bien fondé

3.4 Elimination des paramètres lorsque $\mathcal{A} = T(F)$

Nous nous restreignons dans cette section et dans toute la suite de ce chapitre au cas où $\mathcal{A} = T(F)$. Ce cas revêt une importance particulière, d'une part parce qu'il est plus difficile que la résolution dans $T(F, X)$ (cf [Kun87,Mah88a]), d'autre part parce que c'est ce cas qui est utilisé dans la plupart de nos applications.

Nous envisagerons les cas $\mathcal{A} = RT(F)$ et $\mathcal{A} = T(F, X)$ dans le prochain chapitre. Le cas $\mathcal{A} = T(F)/=_E$ est étudié dans le chapitre 7.

Dans cette section, $\mathcal{F}_I$ sera l'ensemble de tous les (représentants de) problèmes équationnels et $\mathcal{F}_R$ l'ensemble des problèmes sans paramètre. Comme il a été précisé dans la section précédente, nous choisirons pour $\mathcal{F}_{RS}$ l'ensemble des formes irréductibles d'un problème pour le système $\mathcal{R}_0$ que nous allons définir ci-dessous.

3.4.1 Définition de $\mathcal{R}_0$

Ce système de règles de transformation est constitué (d'une partie) des règles données dans la section 3.1 auxquelles a été ajouté un contrôle. Ces règles sont décites, avec leur contrôle, dans les figures 3.7 et 3.8. Il faut noter que certaines d'entre elles ne sont pas nécessaires (par exemple (F_2)) et, de façon plus générale, le contrôle aurait pu être plus "restrictif". Nous avons néanmoins choisi de donner un contrôle relativement général, car la preuve de terminaison reste alors valide pour toute spécialisation. On pourra ainsi raffiner le contrôle en fonction des problèmes étudiés pour obtenir une efficacité maximum, sans avoir à faire une nouvelle preuve de terminaison.

Dans la figure 3.7 nous n'avons fait que reproduire certaines règles de la section 3.1 sans rien ajouter. Par contre, certaines conditions d'application ont été ajoutées dans la figure 3.8. Dans l'expression de ces conditions, nous utilisons la notion de *paramètre résolu* . Un paramètre y est résolu dans une disjonction d'équations et de diséquations d s'il existe une diséquation $y \neq u$ dans d et si y n'a qu'une occurrence dans d.

Nous utilisons aussi la fonction taille-parametre(t) qui dénote la somme des tailles des positions des paramètres dans t. Par exemple, taille-parametre($f(y_1, g(y_1), g(g(y_2)))$)= 6 si y_1 et y_2 sont tous deux des paramètres.

Théoreme 3.16 *Soit $\mathcal{A} = T(F)$. L'application non déterministe des règles données dans les figures 3.7 et 3.8 termine. De plus, les formes irréductibles sont des problèmes sans paramètre.*

Avant de commencer la preuve, donnons tout de suite le corollaire, qui résulte de la remarque fondamentale de la section précédente.

Corollaire 3.17 *Les règles de transformation des figures 3.7 et 3.8 sont complètes par rapport à - $\mathcal{F}_I$ ensemble de tous les problèmes équationnels - $\mathcal{F}_R$ ensemble des problèmes sans paramètres - $\mathcal{A} = T(F)$.*

preuve du théorème 3.16

Preuve de terminaison Nous construisons un certain nombre de fonctions d'interprétation dont la décroissance par application des règles permettra de prouver la terminaison:

- Si d est une disjonction d'équations et de diséquations, $\phi_1(d)$ désigne le nombre de paramètres distincts ayant au moins une occurrence dans d.
- Etant donné une disjonction d'équations et de diséquations $d \equiv e_1 \vee \ldots \vee e_n$, $\phi_2(d)$ désigne le multi-ensemble $\{TM(e_1), \ldots, TM(e_n)\}$ où $TM(e)$ est défini par:
  - $TM(e) = 0$ si l'un des membres de e est un paramètre résolu
  - Sinon, $TM(s = t) = TM(s \neq t) =$ max(taille-param(s),taille-param(t)).

  Par exemple, $\phi_2(y_1 \neq f(g(g(g(y_3))), a) \vee y_3 \neq g(y_2) \vee g(y_4) = g(g(y_5))) = \{0, 2, 3\}$ si les y_i sont des paramètres.
- Si, à nouveau, d est une disjonction d'équations et de diséquations, $\phi_3(d)$ est le nombre d'équations et de diséquations de d dont un des membres est une variable.
- Si $\mathcal{P} \equiv \exists \vec{w}, \forall \vec{y} : d_1 \wedge \ldots \wedge d_n$ est un problème en forme normale conjonctive, $\psi_1(\mathcal{P})$ est le multi-ensemble de triplets

$$\{(\phi_1(d_1), \phi_2(d_1), \phi_3(d_1)), \ldots, (\phi_1(d_n), \phi_2(d_n), \phi_3(d_n))\}$$

- Si $\mathcal{P}$ est un problème équationnel en forme normale conjonctive, $\psi_2(\mathcal{P})$ est la taille totale de $\mathcal{P}$, c'est-à-dire le nombre total de symboles de $F \cup X$ apparaissant dans $\mathcal{P}$.

Elimination des paramètres (Ebf P)

$$(EP_1) \quad \forall \vec{y}, y : P \mapsto \forall \vec{y} : P \qquad \text{Si } y \notin Var(P)$$
$$(EP_2) \quad \forall \vec{y} : P \wedge (y \neq t \vee d) \mapsto \forall \vec{y} : P \wedge d\{y \to t\} \qquad \text{Si } d \text{ est une disjonction d'équations et de diséquations, } y \in \vec{y} \text{ et } y \notin Var(t)$$

$$(EP_3) \quad \vec{y} : P \wedge (z_1 = u_1 \vee \ldots \vee z_n = u_n \vee R) \mapsto \vec{y} : P \wedge R$$

Si

1. Pour tout i, z_i est une variable et est distincte de u_i
2. Pour tout i, $z_i = u_i$ contient au moins une occurrence de paramètre
3. Pour tout i et tout paramètre $y \in Var(z_i, u_i)$, y est infinitaire
4. R ne contient pas d'occurrence de paramètre

$$(EP_4) \quad \forall \vec{y} : P \wedge Q \mapsto \forall \vec{y} : P \wedge Q\{y \to t_1\} \wedge \ldots \wedge Q\{y \to t_n\}$$

Si y est un paramètre de sorte $\underline{s}$ dont le support dans $T(F)$ est $\{t_1, \ldots, t_n\}$.

Elimination des équations et diséquations triviales (T)

$$(T_1) \quad s = s \mapsto \top$$
$$(T_2) \quad s \neq s \mapsto \bot$$

Incompatibilités (I)

$$(I_1) \quad f(t_1, \ldots, t_n) = g(u_1, \ldots, u_m) \mapsto \bot \qquad \text{Si } f \neq g$$
$$(I_2) \quad f(t_1, \ldots, t_n) \neq g(u_1, \ldots, u_m) \mapsto \top \qquad \text{Si } f \neq g$$

Tests d'occurrence (O)

$$(O_1) \quad z = t \mapsto \bot \qquad \text{Si } z \in Var(t)$$
$$(O_2) \quad z \neq t \mapsto \top \qquad \text{Si } z \in Var(t)$$

Figure 3.7: Regles permettant l'élimination des paramètres dans les théories libres

Fusions (F)

$$
\begin{array}{lrcl}
(F_1) & z = t \wedge z = u & \mapsto & z = t \wedge t = u \\
(F_3) & z = t \wedge z \neq u & \mapsto & z = t \wedge t \neq u \\
(F_1') & z = t \wedge (z = u \vee d) & \mapsto & z = t \wedge (t = u \vee d) \\
(F_3') & z = t \wedge (z \neq u \vee d) & \mapsto & z = t \wedge (t \neq u \vee d)
\end{array}
$$

Pour ces règles de fusion, on supposera que:

1. z est une inconnue et pas t
2. t ne contient pas d'occurrence de paramètre
3. u contient une occurrence de paramètre et n'est pas lui-même un paramètre

$$
\begin{array}{lrcl}
(F_2) & z \neq t \vee z \neq u & \mapsto & z \neq t \vee t \neq u \\
(F_4) & z = u \vee z \neq t & \mapsto & u = t \vee z \neq t
\end{array}
$$

Pour ces règles de fusion, on supposera que:

1. z est une variable et t n'est pas une variable
2. u contient une occurrence de paramètre
3. Ou bien taille-param(t) $\leq$ taille-param(u) ou bien u est un paramètre résolu.

Décompositions (D)

$$
\begin{array}{lrcl}
(D_1) & f(t_1,\ldots,t_n) = f(u_1,\ldots u_n) & \mapsto & t_1 = u_1 \wedge \ldots \wedge t_n = u_n \\
(D_2) & f(t_1,\ldots,t_n) \neq f(u_1,\ldots u_n) & \mapsto & t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \\
(D_3) & f(t_1,\ldots,t_n) = f(u_1,\ldots,u_n) \vee d & \mapsto & (t_1 = u_1 \vee d) \wedge \ldots \wedge (t_n = u_n \vee d)
\end{array}
$$

Pour les règles de décomposition, on supposera que $f(t_1,\ldots,t_n)$ ou $f(u_1,\ldots,u_n)$ contient au moins une occurrence de paramètre.

Explosion (E)

$$(Ex_1) \quad \forall \vec{y}: P \mapsto \exists w_1,\ldots,w_p, \forall \vec{y}: P \wedge x = f(w_1,\ldots,w_p)$$

Cette règle ne sera appliquée que si

1. x est une inconnue et $\vec{w} \cap (Var(P) \cup \vec{y} \cup \mathcal{I}) = \emptyset$ et $f \in F$
2. Il existe une équation $x = u$ (ou une diséquation $x \neq u$) dans P telle que u n'est pas une variable et contient au moins une occurrence de paramètre.
3. Aucune des règles **E**bf P, **F**, **D**, **I**, **O**, **T** ne peut s'appliquer.

Figure 3.8: Elimination des paramètres (suite)

	ψ_1	ϕ_1	ϕ_2	ϕ_3	ψ_2
$(F_1),(F_3),(F_1'),(F_3')$	$<_{(1)}$	$=_{(1)}$	$=_{(1)}$	$<_{(1)}$	
$(F_2),(F_4)$	$<$	$=$	$=_{(2)}$	$<$	
$(EP_2),(EP_3),(EP_4)$	$<$	$<$			
D	$<$	$=$	$<_{(3)}$		
I, T, O, (EP_1)	$\leq$	$\leq$	$\leq$	$\leq$	$<$

Figure 3.9: Monotonie des fonctions d'interprétation

Nous prouvons tout d'abord que la fonction $\Phi = (\psi_1, \psi_2)$ est strictement décroisante par application d'une règle quelconque, distincte de (Ex_1). Comme le codomaine de Φ est obtenu par compositions d'extensions lexicographiques et multi-ensembles à partir des entiers naturels, cela prouvera la terminaison des transformations, lorsque l'explosion n'est pas prise en considération.

Nous prouverons ensuite que, si $\mathcal{P} \mapsto_{Ex} \mathcal{P}'$, alors, pour tout $\mathcal{P}''$ tel que $\mathcal{P}' \mapsto_{\mathcal{R}_0} \mathcal{P}''$, $\Phi(\mathcal{P}'') < \Phi(\mathcal{P})$. Cela achèvera la preuve de terminaison puisque, s'il existait une chaine infinie de transformation, nous pourrions en extraire une chaine infinie strictement décroissante pour Φ, ce qui est absurde.

Terminaison des règles lorsqu'on ne considère pas (Ex_1) Le tableau de la figure 3.9 résume les variations des fonctions $\psi_1, \psi_2, \phi_1, \phi_2, \phi_3$ par application des règles. On trouve à l'intersection de la ligne R et de la colonne ϕ_i l'un des symboles $=, \leq, <$ correspondant au sens de variations de ϕ_i lorsque R est appliquée. Pour chaque résultat non trivial nous indiquons un renvoi à une explication plus détaillée.

(1) $(F_1),(F_3),(F_1'),(F_3')$ **font strictement décroitre** ψ_1
C'est une conséquence du contrôle : t ne contient pas de paramètre. Par conséquent, les fonctions ϕ_1 et ϕ_2 restent inchangées par application de ces règles. Par ailleurs, z est une variable et pas u. Il en résulte que ϕ_3 est strictement décroissante pour une certaine disjonction d'équations et de diséquations.

(2) $(F_2),(F_4)$ **ne modifient pas** ϕ_2
C'est une conséquence d'à la fois la définition de $TM(e)$ et du contrôle. En effet, la seule chose qui est modifiée par les règles (F_2) et (F_4) est l'équation $z = u$ (resp. la diséquation $z \neq u$) qui est transformée en $t = u$ (resp. $t \neq u$). Dans les deux cas, z ne peut être un paramètre résolu puisqu'il a au moins deux occurrences. De plus,

- ou bien taille-param$(t) \leq$ taille-param(u) et, dans ce cas, $TM(z = u)$ = taille-param$(u) = TM(t = u)$ (resp. $TM(z \neq u) = TM(t \neq u)$)
- ou bien u est un paramètre résolu. Dans ce dernier cas, $TM(z = u) = TM(t = u) = 0$ (resp. $TM(z \neq u) = TM(t \neq u) = 0$).

(3) Les règles de décomposition font strictement décroitre ψ_1

Supposons que $\mathcal{P} \mapsto_{D_1} \mathcal{P}'$. Soit alors

$$\psi_1(\mathcal{P}) = \{(a_1, b_1, c_1), \ldots, (a_n, b_n, c_n), (a, b, c)\}$$

où $a = \phi_1(f(t_1, \ldots, t_m) = f(u_1, \ldots, u_m))$,
$b = \{\max(\text{taille-param}(f(t_1, \ldots, t_m)), \text{taille-param}(f(u_1, \ldots, u_m)))\}$ et c n'a pas d'importance. On peut alors écrire:

$$\psi_1(\mathcal{P}') = \{(a_1, b_1, c_1), \ldots, (a_n, b_n, c_n), (a'_1, b'_1, c'_1), \ldots, (a'_m, b'_m, c'_m)\}$$

où $a'_i = \max(\text{taille-param}(t_i), \text{taille-param}(u_i))$. Mais, pour tout indice i, $a'_i \leq a$ et $b'_i < b$, puisque, comme le contrôle l'impose, $f(t_1, \ldots, t_m) = f(u_1, \ldots, u_m)$ contient au moins une occurrence de paramètre. Cela signifie que $\{(a, b, c)\} > \{(a'_1, b'_1, c'_1), \ldots, (a'_m, b'_m, c'_m)\}$. Par conséquent ψ_1 est strictement décroissante.

Supposons maintenant que $\mathcal{P} \mapsto_{D_2} \mathcal{P}'$. Soit

$$\psi_1(\mathcal{P}) = \{d_1, \ldots, d_n, (a, \{b_1, \ldots, b_k, TM(f(t_1, \ldots, t_m) \neq f(u_1, \ldots, u_m))\}, c)\}$$

Alors,

$$\psi_1(\mathcal{P}') = \{d_1, \ldots, d_n, (a, \{b_1, \ldots, b_k, TM(t_1 \neq u_1), \ldots, TM(t_m \neq u_m)\}, c')\}$$

Chaque $TM(t_i \neq u_i)$ est strictement plus petit que $TM(f(t_1, \ldots, t_m) \neq f(u_1, \ldots, u_m))$ puisque ou bien $f(t_1, \ldots, t_m)$ ou bien $f(u_1, \ldots u_m)$ contient une occurrence de paramètre, d'après le contrôle. Par suite, ψ_1 est à nouveau strictement décroissante.

Supposons enfin que $\mathcal{P} \mapsto_{D_3} \mathcal{P}'$. Soit:

$$\psi_1(\mathcal{P}) = \{d_1, \ldots, d_n, (a, \{b_1, \ldots, b_k, TM(f(t_1, \ldots, t_m) = f(u_1, \ldots, u_m))\}, c)\}$$

Alors,

$$\psi_1(\mathcal{P}') = \{d_1, \ldots, d_n, (a_1, \{b_1, \ldots, b_k, TM(t_1 = u_1)\}, c'_1), \ldots, (a_m, \{b_1, \ldots, b_k, TM(t_m = u_m)\}, c'_m)\}$$

Mais, pour tout indice i, $a_i \leq a$ et

$$TM(t_i = u_i) < TM(f(t_1, \ldots, t_m) = f(u_1, \ldots, u_m))$$

(à cause du contrôle). Par conséquent, ψ_1 est à nouveau décroissante.

Cas de la règle (Ex_1)

Supposons que $\mathcal{P} \mapsto_{Ex_1} \mathcal{P}'$ et que $\psi_1(\mathcal{P}) = \{d_1, \ldots, d_n\}$. Alors, $\psi_1(\mathcal{P}') = \{d_1, \ldots, d_n, (0, \{0\}, 1)\}$. Nous voulons alors prouver que, pour tout $\mathcal{P}''$ tel que $\mathcal{P}' \mapsto_{\mathcal{R}_0} \mathcal{P}''$, $\Phi(\mathcal{P}'') < \Phi(\mathcal{P})$.

A cause du contrôle imposé à l'explosion, les règles **I**, **O**, **T**, **D**, **E**bf P ne s'appliquent pas à $\mathcal{P}$. Par conséquent elles ne s'appliquent pas à $\mathcal{P}'$. D'autre part, une règle de fusion s'applique à $\mathcal{P}'$ puisque x est supposé apparaitre dans

une équation $x = t$ (ou une diséquation $x \neq t$), t contenant une occurrence de paramètre. A cause du contrôle imposé à (Ex_1), on ne peut ainsi pas appliquer cette règle à $\mathcal{P}'$. Finalement, la transformation $\mathcal{P}' \mapsto \mathcal{P}''$ utilise nécessairement une règle de fusion entre l'équation $x = f(w_1, \ldots, w_p)$ et $x = u$ (ou $x \neq u$) où u n'est pas un paramètre et contient une occurrence de paramètre. (Cf le contrôle des règles de fusion).

Cela signifie que $\psi_1(\mathcal{P}'') = \{d_1, \ldots, d_{n-1}, d', (0, \{0\}, 1)\}$ où $d_n = (a_1, a_2, a_3)$, $d' = (a_1, a_2, a_3 - 1)$ et $a_1 \geq 1$. Comme $d_n > d'$ et $d_n > (0, \{0\}, 1)$, on obtient bien $\psi_1(\mathcal{P}'') < \psi_1(\mathcal{P})$.

Preuve de complétude

Nous devons ici prouver que tout problème équationnel irréductible pour les règles des figures 3.7 et 3.8 ne contient pas de paramètre. Il nous suffit donc d'envisager tous les cas d'occurrence de paramètre dans un problème et de montrer que, dans chaque cas, l'une des règles mentionnées s'applique.

Un paramètre apparait dans une équation ou une diséquation entre deux termes non variable
Dans ce cas, une des règles de **D**, **I** ou **T** s'applique.

Un paramètre apparait dans une équation ou une diséquation dont l'un des membres est une inconnue el l'autre n'est pas une variable.
Alors, si aucune autre règle ne s'applique, (Ex_1) s'applique.

Un paramètre y est membre d'une diséquation
On peut alors appliquer l'une des règles $(EP_2), (T_2), (O_2)$.

Autres cas d'occurrence de paramètres dans une équation
Toute équation (ou diséquation) contenant une occurrence de paramètre doit avoir un paramètre pour un de ses membres, sinon nous sommes dans l'un des cas qui précèdent. On peut alors appliquer l'une des règles $(T_1), (O_1), (EP_4), (EP_5)$.

Un paramètre apparait dans l'en tête du problème
Si nous ne sommes dans aucun des cas qui précèdent, c'est que la règle (EP_1) est applicable.

□

Remarques

- Les règles de fusion (F_2) et (F_4) ne sont pas utilisées dans la preuve de complétude. Le théorème est donc toujours vrai si l'on ne considère pas ces règles. Mais inversement, la terminaison ayant été prouvée "malgré" ces règles, celles-ci peuvent être utilisées pour améliorer l'efficacité.

- Les tests d'occurrence sont effectivement utilisés dans la preuve de complétude. Celle-ci n'est donc plus valide lorsque $\mathcal{A} = RT(F)$.

3.5 Définitions contraintes lorsque $\mathcal{A} = T(F)$

Nous nous intéressons maintenant à la poursuite de la "simplification" envisagée dans la section précédente: l'ensemble des problèmes initiaux $\mathcal{F}_I$ est constitué des problèmes sans paramètre, et l'ensemble $\mathcal{F}_R$ des formes résolues est l'ensemble des définitions contraintes telles qu'elles ont été définies dans la section 3.3. Rappelons en ici la définition:
Lorsque $\mathcal{A}$ est une sous-algèbre de $T(F, X)$, un problème équationnel $\mathcal{P}$ est dit être une *définition contrainte* (par rapport à $\mathcal{A}$ et $\mathcal{I}$) si $\mathcal{P}$ est ou bien $\top$ ou $\bot$ ou bien une conjonction d'équations et de diséquations $\exists w_1, \ldots, w_k : x_1 = t_1 \wedge \ldots \wedge x_m = t_m \wedge x'_1 \neq u_1 \wedge \ldots \wedge x'_p \neq u_p$ où

1. $x_1, \ldots, x_m$ sont des variables et n'apparaissent qu'une fois dans $\mathcal{P}$
2. pour tout indice $1 \leq i \leq p$, x'_i est infinitaire et est distinct de u_i

La proposition 3.13 nous assure aussi qu'une définition contrainte distincte de $\bot$ a au moins une solution dans $T(F)$.

Le système de règles $\mathcal{R}_1$ permet ainsi de transformer tout problème sans paramètre en un ensemble fini de définitions contraintes et, par conséquent, de décider de l'existence d'une solution dans $T(F)$ d'un problème équationnel, en utilisant les résultats de la section précédente[11].

3.5.1 Définition de $\mathcal{R}_1$

Comme $\mathcal{R}_0$, $\mathcal{R}_1$ est constitué des règles des figures 3.1, 3.2, 3.3 auxquelles ont impose un certain nombre de conditions supplémentaires. (Ces conditions sont différentes de celles que nous avons imposées pour $\mathcal{R}_0$). De même que dans la section précédente, ce contrôle est "le plus libéral possible". Nous donnons ainsi, par exemple, à la fois les règles de fusion et les règles de remplacement, alors que ces dernières suffiraient. Par différentes spécialisations du contrôle on retrouve ainsi différents algorithmes d'unification connus (comme ceux de Herbrand [Her30], Martelli-Montanari [MM82] ou de Colmerauer [Col84]).

Les règles de $\mathcal{R}_1$ sont résumées dans les figures 3.10 et 3.11. Nous utilisons pour donner le contrôle certaines notions :

- La *taille* d'un terme est le nombre de ses noeuds (lorsqu'il est considéré comme un arbre)
- Une variable z est *presque résolue* dans un problème $\mathcal{P} \equiv d_1 \wedge \ldots \wedge d_n$ en forme normale conjonctive si l'un des d_i est de la forme $z = t$ où t est un terme distinct de z.

[11] En fait, il n'est pas nécessaire de passer par l'étape "problèmes sans paramètre" pour obtenir des définitions contraintes. Nous verrons en effet dans la section suivante comment combiner les transformations (autrement qu'en séquence). Néanmoins cet enchainement naturel permet plus de clarté dans l'exposé. Nous verrons aussi dans la section 3.7 que la deuxième partie (celle qui est décrite dans cette section) peut n'être utilisée qu'une fois dans la simplification des formules équationnelles alors que l'élimination des paramètres doit l'être plusieurs fois. La distinction peut ainsi permettre certaines optimisations

Elimination des équations et diséquations triviales (T)

$$(T_1) \quad s = s \mapsto \top$$
$$(T_2) \quad s \neq s \mapsto \bot$$

Remplacements (R)

$$(R_1) \quad z = t \wedge P \mapsto z = t \wedge P\{z \rightarrow t\}$$

Le remplacement n'est effectué que si z est une variable, $z \notin Var(t)$, $z \in Var(P)$, t ne contient pas de paramètre et -ou bien t n'est pas une variable -ou bien t a une occurrence dans P.

Fusions (F)

$$(F_1) \quad z = t \wedge z = u \mapsto z = t \wedge t = u$$
$$(F_2) \quad s \neq t \vee s \neq u \mapsto s \neq t \vee t \neq u$$
$$(F_3) \quad z = t \wedge z \neq u \mapsto z = t \wedge t \neq u$$
$$(F_4) \quad s = t \vee s \neq u \mapsto u = t \vee s \neq u$$
$$(F'_1) \quad z = t \wedge (z = u \vee d) \mapsto z = t \wedge (t = u \vee d)$$
$$(F'_3) \quad z = t \wedge (z \neq u \vee d) \mapsto z = t \wedge (t \neq u \vee d)$$

Pour ces règles de fusion, on supposera que z est une variable, t n'est pas une variable et - ou bien taille(t) $\leq$ taille(u) -ou bien u est une variable résolue.

Incompatibilités (I)

$$(I_1) \quad f(t_1, \ldots, t_n) = g(u_1, \ldots, u_m) \mapsto \bot \quad \text{Si } f \neq g$$
$$(I_2) \quad f(t_1, \ldots, t_n) \neq g(u_1, \ldots, u_m) \mapsto \top \quad \text{Si } f \neq g$$

Figure 3.10: Règles permettant la transformation en définitions contraintes

- Une variable est *résolue* dans un problème équationnel $\mathcal{P}$ si elle est presque résolue et n'a qu'une occurrence dans $\mathcal{P}$.

Théoreme 3.18 *Soit $\mathcal{A} \subseteq T(F, X)$. L'application non déterministe des règles données dans les figures 3.10 et 3.11 à un problème sans paramètre termine toujours. De plus, les problèmes irréductibles sont des définitions contraintes.*

Une conséquence directe de ce théorème est le résultat de complétude suivant:

Corollaire 3.19 *Le système $\mathcal{R}_1$ est complet par rapport à - $\mathcal{F}_I$, ensemble des problèmes sans paramètre - $\mathcal{F}_R$, ensemble des définitions contraintes - $\mathcal{A} \subseteq T(F, X)$.*

Preuve du théorème 3.18

Décompositions (D)

$$
\begin{array}{lrcl}
(D_1) & f(t_1,\ldots,t_n) = f(u_1,\ldots u_n) & \mapsto & t_1 = u_1 \wedge \ldots \wedge t_n = u_n \\
(D_2) & f(t_1,\ldots,t_n) \neq f(u_1,\ldots u_n) & \mapsto & t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \\
(D_3) & f(t_1,\ldots,t_n) = f(u_1,\ldots,u_n) \vee d & \mapsto & (t_1 = u_1 \vee d) \wedge \ldots \wedge (t_n = u_n \vee d)
\end{array}
$$

Tests d'occurrence (O)

$$
\begin{array}{lrcll}
(O_1) & z = t & \mapsto & \bot & \text{Si } z \in Var(t) \\
(O_2) & z \neq t & \mapsto & \top & \text{Si } z \in Var(t)
\end{array}
$$

Explosion (E)

$$(Ex_2) \quad P[x \neq t] \mapsto P[x \neq t] \wedge x = u$$

Cette règle n'est utilisée que si

1. Le support de $sort(x)$ dans $\mathcal{A} \subseteq T(F,X)$ est fini. u est un des éléments de ce support
2. **M, O, R, I, D** ne s'appliquent pas

Choix non déterministe (NC)

$$(Nc) \quad \forall \vec{y}\colon P \wedge (P_1 \vee P_2) \mapsto \forall \vec{y}\colon P \wedge P_1 \qquad \text{Si } Var(P_1) \cap \vec{y} = \emptyset \text{ ou } Var(P_2) \cap \vec{y} = \emptyset$$

Figure 3.11: Règles pour les définitions contraintes (suite)

	ϕ_1	ϕ_2	ϕ_3	ϕ_4
R	$=_{(1)}$	$<_{(2)}$		
F	$=_{(1)}$	$\leq_{(3)}$	$\leq_{(4)}$	$<$
T, I, O, (Nc)	$\leq_{(1)}$	$\leq_{(5)}$	$<_{(5)}$	
D	$\leq_{(1)}$	$\leq_{(6)}$	$<_{(7)}$	
E	$<_{(8)}$			

Figure 3.12: Variations des fonctions d'interprétations suivant la règle appliquée

Preuve de terminaison Comme précédemment, nous allons donner un certain nombre de fonctions d'interprétation dont la décroissance permettra de prouver la terminaison.

- $\phi_1(\mathcal{P})$ est le nombre de variables de $\mathcal{P}$ qui ne sont pas presque résolues
- $\phi_2(\mathcal{P})$ est le nombre de variables de $\mathcal{P}$ qui ne sont pas résolues
- Si e est une équation ou une diséquation, $Tm(e)$ est égal à 0 si l'un des membres de e est une variable résolue et au maximum des des tailles de ses deux membres sinon.
- Si $d \equiv e_1 \vee \ldots \vee e_m$ est une disjonction d'équations et de diséquations, $M(d)$ est le multi-ensemble $\{Tm(e_1), \ldots, Tm(e_n)\}$.
- $\phi_3(d_1 \wedge \ldots \wedge d_n)$ où $d_1, \ldots, d_n$ sont des disjonctions d'équations et de diséquations est le multi-ensemble $\{M(d_1), \ldots, M(d_n)\}$.
- $\phi_4(\mathcal{P})$ est le nombre total d'occurrences dans $\mathcal{P}$ d'une variable comme membre d'une équation ou d'une diséquation.

Soit alors $\Phi = (\phi_1, \phi_2, \phi_3, \phi_4)$. Les variations de Φ sont résumées dans le tableau de la figure 3.12.

(1) ϕ_1 est toujours croissante au sens large

Soit $x = t$ une équation de $\mathcal{P}$ qui n'est pas à l'intérieur d'une disjonction et telle que x est une variable et $x \not\equiv t$. (Autrement dit, x est une variable presque résolue). ϕ_1 ne peut croitre que si une transformation de $\mathcal{P}$ conduit à la disparition d'une variable presque résolue. Il nous suffit ainsi de montrer que x reste presque résolue après une transformation quelconque sur $\mathcal{P}$.

L'élimination ou la modification de $x = t$ ne peut être obtenue que par application d'une des règles (F_1) ou (R_1) ou par effet de bord de (T_2) ou (O_1) (i.e. par normalisation du problème obtenu après une telle transformation). Dans ce dernier cas, le problème obtenu est $\perp$ et donc ϕ_1 est trivialement décroissante. Il reste ainsi trois cas à envisager:

1. (R_1) ou (M_1) transforme l'équation $x = t$ en une équation $t = u$. Une telle transformation n'est possible que s'il existe dans $\mathcal{P}$ une équation $x = u$

qui, elle non plus, n'est pas à l'intérieur d'une disjonction. Par conséquent, après application de la règle, x est toujours presque résolue.

2. (R_1) ou (M_1) transforme $x = t$ en $x = x$. Il faudrait avoir dans ce cas une autre occurrence de $x = t$, ce qui est en contradiction avec le fait que $\mathcal{P}$ est en forme normale conjonctive.
3. (R_1) ou (M_1) transforme $x = t$ en une équation $x = u$ qui est déjà dans $\mathcal{P}$. Dans ce cas, la normalisation va éliminer $x = u$. Mais -ou bien t n'est pas une variable -ou bien t apparait comme membre d'une autre équation de $\mathcal{P}$. dans les deux cas, le nombre de variables presque résolues reste inchangé.

(2) ϕ_2 est strictement décroissante par application de (R_1)
A cause du contrôle imposé à (R_1), z n'est pas une variable résolue du problème auquel (R_1) est appliquée (puisque z doit avoir au moins une occurrence dans P). Par contre, z est résolue après application du remplacement puisque t ne contient pas d'occurrence de z. Enfin, les variables résolues avant application de la règle le sont aussi après puisque (toujours à cause du contrôle) t ne peut être une variable résolue. Il en résulte que, par remplacement, le nombre de variables résolues a décrû de 1.

(3) Les règles de fusion font décroitre (au sens large) ϕ_2
Ces règles n'introduisent pas de nouvelles variables. A cause du contrôle, elles ne peuvent pas non plus en dupliquer une qui n'a qu'une occurrence. Par conséquent elles ne peuvent faire décroitre le nombre de variables résolues.

(4) Les fusions font décroitre (au sens large) ϕ_3
C'est une conséquence du contrôle:

- Ou bien taille$(t) \leq$ taille(u) et, par définition, ϕ_3 n'est pas modifiée par la fusion
- Ou bien u est une variable résolue. Dans ce cas,

$$Tm(z = u) = Tm(t = u) = Tm(z \neq u) = Tm(t \neq u) = 0$$

et ϕ_3 est donc inchangée

(5) T, I, O, (Nc) font strictement décroitre ϕ_3
C'est immédiat si l'on se rappelle que le résultat d'une transformation est immédiatement réduit à sa forme normale conjonctive.

(6) Les décompositions font décroitre (au sens large) ϕ_2
En effet, la règle (D_3) ne peut dupliquer que des variables non résolues.

(7) Les décompositions font strictement décroitre ϕ_3
Soit $n = Tm(e)$ où e est une équation (ou une diséquation) à laquelle on applique une règle de décomposition. Par dédfinition, $n > 0$.

- Si $\mathcal{P} \mapsto_{D_1} \mathcal{P}'$, alors

$$\phi_3(\mathcal{P}) = \{\{n\}, a_1, \ldots, a_k\}$$

et

$$\phi_3(\mathcal{P}') = \{\{n_1\}, \ldots, \{n_m\}, a_1, \ldots, a_k\}$$

où $n_i < n$ pour tout i. (Par définition de Tm). Donc $\phi_3(\mathcal{P}') < \phi_3(\mathcal{P})$.

- Si $\mathcal{P} \mapsto_{D_2} \mathcal{P}'$, alors

$$\phi_3(\mathcal{P}) = \{\{n, b_1, \ldots, b_l\}, a_1, \ldots, a_k\}$$

et

$$\phi_3(\mathcal{P}') = \{\{n_1, \ldots, n_m, b_1, \ldots, b_l\}, a_1, \ldots, a_k\}$$

où $n_i < n$ pour tout i. On a donc encore $\phi_3(\mathcal{P}') < \phi_3(\mathcal{P})$.

- Si $\mathcal{P} \mapsto_{D_3} \mathcal{P}'$, alors

$$\phi_3(\mathcal{P}) = \{\{n, b_1, \ldots, b_l\}, a_1, \ldots, a_k\}$$

et

$$\phi_3(\mathcal{P}') = \{\{n_1, b_1, \ldots, b_l\}, \ldots, \{n_m, b_1, \ldots, b_l\}, a_1, \ldots, a_k\}$$

où $n_i < n$ pour tout i. On a encore $\phi_3(\mathcal{P}') < \phi_3(\mathcal{P})$.

(8) **ϕ_1 est strictement décroissante par explosion.**
En effet, s'il est possible d'exploser x, x est membre d'une diséquation $x \neq u$. De plus, les règles **R** et **I** ne peuvent s'appliquer. Par conséquent, le problème $\mathcal{P}$ auquel s'applique la règle n'est pas de la forme $\exists \vec{w}, \forall \vec{y} : x = t \wedge P$. Autrement dit, x n'est pas presque résolue. Comme la règle d'explosion ajoute l'équation $x = v$ où v ne contient pas de variable, cela implique la décroissance stricte de ϕ_1.[12]

Comme la composée lexicographique et l'extension multi-ensemble d'ordres bien fondés sont bien fondées, il ne peut y avoir de suite infinie décroisante de la forme $\Phi(\mathcal{P}_i)$. Comme Φ est strictement décroissante par application des règles, nous pouvons conclure que le système $\mathcal{R}_1$ est à terminaison finie.

Preuve de complétude Nous prouvons ici que n'importe quel problème sans paramètre qui n'est pas une définition contrainte est réductible par $\mathcal{R}_1$. Soit $\mathcal{P}$ un tel problème.

Si $\mathcal{P}$ contient des disjonctions On peut appliquer (Nc).

Si $\mathcal{P}$ contient une équation ou une diséquation dont les membres ne sont pas des variables
On peut appliquer une règle de décomposition ou d'incompatibilité.

Si une variable x d'une équation $x = t$ apparait deux fois dans $\mathcal{P}$ et que, si t est une variable, alors t apparait aussi deux fois dans $\mathcal{P}$. Alors -ou bien $x \in Var(t)$ et l'on peut appliquer (O_1) -ou bien il est possible d'appliquer (R_1).

Si une variable x d'une diséquation $x = t$ a une sorte à support fini dans $\mathcal{A}$
Alors, si aucune autre règle ne s'applique, on peut utiliser l'explosion.

S'il y a une équation ou une diséquation triviale alors **T** s'applique.

Tous les cas sont maintenant couverts, ce qui achève la preuve.

[12] Remarquons que nous aurions pu faire le même raisonnement avec les règles de fusion à la place du remplacement.

□

Remarques

1. Beaucoup de règles énoncées ici sont inutiles (ou redondantes). Prenons deux exemples.

   (a) Comme les problèmes étudiés ne contiennent pas de paramètre, on peut appliquer systématiquement la règle (Nc). Nous n'avons alors à considérer que des problèmes sans disjonctions. Les règles telles que $(F_2), (D_3), \ldots$ sont alors inutiles. Si nous les avons maintenues, c'est qu'il est important de pouvoir "repousser" l'application de (Nc). En fait, la preuve de terminaison nous permet de n'aspppliquer (Nc) qu'en dernier lieu. Nous utiliserons effectivement cette possibilité dans le chapitre suivant pour obtenir des problèmes d'unification.

   (b) Les règles de fusion sont inutiles puisque le remplacement couvre tous les cas utiles. Mais si l'on veut, par exemple, résoudre les problèmes dans les arbres rationnels ou dans les algèbres avec sortes ordonnés, alors son utilisation doit être restreinte. Comme notre preuve de terminaison ne fait aucune hypothèse sur l'utilisation respective des fusions et des remplacements, ceux-ci peuvent être utilisés avec n'importe quelle restrictions, la terminaison reste assurée. C'est ce que nous ferons largement dans le chapitre suivant avec la résolution des problèmes équationnels dans les arbres rationnels et dans les algèbres avec sortes ordonnées.

   La place laissée ainsi à diverses utilisations des règles permet donc une preuve "générique" de terminaison. C'est pourquoi nous disions plus haut que, par diverses spécialisations, il est possible de retrouver les algorithmes classiques d'unification. Il faut cependant prendre garde que la preuve de complétude quant à elle utilise la "libéralité" du contrôle. Il faudra donc une nouvelle preuve de complétude pour chaque spécialisation.

2. On peut se demander s'il n'est pas possible de proposer un contrôle encore moins restrictif et permettant néanmoins la peuve de terminaison: Montrons donc que les conditions que nous avons imposées sont bien nécessaires pour prouver la terminaison;

   (a) Si, dans la règle de fusion (F_1), on autorise t à être une variable,

   $$x = y \wedge x = z \mapsto_{F_1} x = y \wedge y = z$$

   ce deuxième problème étant, à renommage près, le problème de départ.

   (b) Si l'on autorise (toujours dans la fusion) à la fois la taille de t à être plus grande que celle de u et u à ne pas être une variable résolue, $x = g(g(z)) \wedge x = g(y) \wedge y = g(g(x)) \wedge z = g(g(y))$

   $$\begin{array}{ll} \mapsto_{F_1} & x = g(g(z)) \wedge g(g(z)) = g(y) \wedge y = g(g(x)) \wedge z = g(g(y)) \\ \mapsto_{D_1} & x = g(g(z)) \wedge y = g(z) \wedge y = g(g(x)) \wedge z = g(g(y)) \\ \mapsto_{F_1} & x = g(g(z)) \wedge g(g(x)) = g(z) \wedge y = g(g(x)) \wedge z = g(g(y)) \\ \mapsto_{D_1} & x = g(g(z)) \wedge z = g(x) \wedge y = g(g(x)) \wedge z = g(g(y)) \end{array}$$

Or ce dernier problème s'obtient à partir du premier par une permutation circulaire des trois variables: $\{x \to z;\ y \to x;\ z \to y\}$.

3. Le deuxième cas d'application de (F_1) (i.e. u est une variable résolue) est indispensable : il permet de résoudre les problèmes d'équations entre variables en évitant le remplacement. (Je remercie JP. Jouannaud qui m'a indiqué cette solution. Elle est maintenant utilisée dans [DJ88]).

3.6 Combinaison des transformations

Il n'est pas, en fait, nécessaire de passer par des problèmes sans paramètres pour obtenir des définitions contraintes. Cette façon de faire introduit un contrôle qui, bien qu'utile pour la clarté, ne sert pas vraiment pour prouver la terminaison. Dans de nombreux cas pratiques il apparait même que ce contrôle est extrêmement coûteux. (Par exemple, si l'on "découvre" une incompatibilité seulement dans la deuxième phase, ce qui est le cas notamment si l'on a une équation comme $f(a) = f(b)$.)

Nous supposerons dans cette section que $\mathcal{A} = T(F)$ (si bien que toutes les règles sont adéquates). $\mathcal{R}_2$ est alors le système de règles obtenu par réunion de $\mathcal{R}_0$ et $\mathcal{R}_1$. Plus précisément, les règles de transformation sont celles qui sont contenues dans l'une des figures 3.7, 3.8, 3.10, 3.11. Si une règle apparait deux fois dans ces figures, une fois avec les conditions $C1$ et une fois avec les conditions $C2$, le contrôle qui lui est associé dans $\mathcal{R}_2$ est "$C1$ ou $C2$". En plus, nous supposerons que

- Pour le remplacement $((R_1))$, t ne contient pas d'occurrence de paramètre
- Pour les fusions $(\mathbf{F})$, t ne contient pas d'occurrence de paramètre
- (Ex_1) n'est employée que si aucune autre règle n'est applicable

Par définition, et à cause des résultats de complétude des sections précédentes, $\mathcal{R}_2$ est complet par rapport à $\mathcal{F}_I$ ensemble de tous les problèmes équationnels, $\mathcal{F}_R$ ensemble des définitions contraintes et $\mathcal{A} = T(F)$. Le contrôle consistant à effectuer d'abord $\mathcal{R}_0$ puis $\mathcal{R}_1$ n'est en effet qu'une spécialisation de $\mathcal{R}_2$.[13] Le seul résultat sérieux à prouver est donc la terminaison.

Théoreme 3.20 *L'application non déterministe des règles de $\mathcal{R}_2$ à n'importe quel problème équationnel termine. Les formes irréductibles sont des définitions contraintes.*

Preuve

Cette preuve se fait en utilisant une combinaison des ordres utilisés dans les deux sections précédentes. Notons en ajoutant un indice 0 les fonctions définies dans la preuve du théorème 3.16 et avec un indice 1 celles qui sont utilisées dans la preuve du théorème 3.18. Par exemple, $\phi_{1,0}$ est la fonction qui associe à une disjonction d'équations et de

[13] Noter que, si l'on libéralise le contrôle, les résultats de complétude sont préservés, mais pas nécessairement la terminaison. À l'inverse, si l'on spécialise le contrôle, les résultats de terminaison sont préservés, mais pas nécessairement la complétude.

	$\phi_{1,0}$	$\phi_{2,0}$	$\phi_{3,2}$	$\phi_{1,1}$	$\phi_{2,1}$	M_1	$\phi_{4,2}$
R	$=_{(1)}$	$=_{(1)}$	$\leq_{(1)}$	$=$	$<$		
F	$=_{(2)}$	$=_{(2)}$	$\leq_{(2)}$	$\leq$	$\leq$	$\leq$	$<$
$\mathbf{T}, \mathbf{I}, \mathbf{O}, (Nc), (EP_1)$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$<$	
$(EP_2), (EP_3), (EP_4)$	$<$						
D	$\leq_{(3)}$	$\leq_{(3)}$	$\leq_{(3)}$	$\leq$	$\leq$	$<$	
(Ex_2)	$=_{(4)}$	$=_{(4)}$	$=_{(4)}$	$<$			

Figure 3.13: Variations des fonctions d'interprétation par application des règles de $\mathcal{R}_2$

diséquations d le nombre de paramètres distincts ayant une occurrence dans d et $\phi_{1,1}(\mathcal{P})$ est le nombre de variables du problème équationnel $\mathcal{P}$ qui ne sont pas presque résolues.

Si $\mathcal{P} \equiv \exists \vec{w}, \forall \vec{y} : d_1 \wedge \ldots \wedge d_n$, $\Phi_2(\mathcal{P})$ désignera le multi-ensemble:

$$\{\psi_2(d_1), \ldots, \psi_2(d_n)\}$$

où $\psi_2(d_i)$ est le 7-uple

$$(\phi_{1,0}(d_i), \phi_{2,0}(d_i), \phi_{3,2}(d_i), \phi_{1,1}(\mathcal{P}), \phi_{2,1}(\mathcal{P}), M_1(d_i), \phi_{4,2}(d_i))$$

où $\phi_{3,2}(d_i)$ désigne le nombre d'équations et de diséquations de d_i dont un des membres est une variable et l'autre est un terme non variable contenant au moins une occurrence de paramètre et $\phi_{4,2}(d_i)$ est le nombre de membres d'équations et de diséquations de d_i qui sont des variables.

Le tableau de la figure 3.13 résume les variations de ces fonctions par application des règles de transformation. Essentiellement, les résultats de décroissance sont obtenus de la même façon que dans les sections précédentes. La seule vérification qu'il reste à faire est qu'aucune des règles ne fait croitre la composée lexicographique des trois premières fonctions. De même que dans la section 3.4, nous n'envisageons pas dans ce tableau la règle d'explosion (Ex_1), qui ne fait pas directement décroitre Φ_2.

(1) Le contrôle interdisant de remplacer x par t lorsque t contient des occurrences de paramètre, $\phi_{1,0}$ et $\phi_{2,0}$ ne sont pas modifiées par application de (R_1). De plus, $\phi_{3,2}$ ne peut pas croitre puisqu'une équation dont aucun des membres n'est une variable ne peut être transformée par (R_1) en une équation dont un des membres est une variable.

(2) Les fusions ne font pas croitre les trois premières fonctions, pour des raisons semblables à celles qui viennent d'être évoquées

(3) C'est là qu'est le plus gros problème. En fait, c'est la règle D_3 qui, permettant éventuellement la duplication de paramètres par décomposition d'une équation qui n'en contient pas, nous a obligé à construire une fonction d'interprétation si compliquée (sans elle, il suffit de prendre la composée lexicographique de $\psi_{1,0}$ et de Φ_1).

On peut tout d'abord éliminer les cas d'application d'une décomposition à une équation (ou une diséquation) contenant des occurrences de paramètre puisque la décroissance du multi-ensemble des couples $(\phi_{1,0}(d_i), \phi_{2,0}(d_i))$ a été prouvée pour le théorème 3.16. Dans les autres cas, (D_1) et (D_3) remplacent une composante du multi-ensemble par un multi-ensemble de 7-uples dont les premières composantes sont identiques et la 6ième a décrû: une certaine disjonction d_i a été transformée en $d'_1 \wedge \ldots \wedge d'_k$ avec

$$\psi_2(d_i) = (a_1, \ldots, a_7)$$

et

$$\psi_2(d'_j) = (a_1, a_2, a_3, a'_{4,j}, a'_{5,j}, a'_{6,j}, a'_{7,j})$$

où, pour tout j, $a'_{4,j} \leq a_4$, $a'_{5,j} \leq a_5$, $a'_{6,j} < a_6$[14]. Ce qui prouve bien la décroissance de Φ_2.

Pour la règle (D_2), on a directement la décroissance de la sixième composante.

(4) (Ex_2) ne modifie pas $\phi_{1,0}, \phi_{2,0}, \phi_{3,0}$ car la règle "ne touche pas à la structure de paramètres". Par contre, la règle ajoute un élément au multi-ensemble. Cependant, elle fait décroitre toutes les quatrièmes composantes (cf preuve du théorème 3.18) ce qui permet d'assurer la décroissance de Φ_2.

Cas de la règle (Ex_1) Comme dans la preuve du théorème 3.16, (Ex_1) ne fait pas décroitre Φ_2 mais, à cause du contrôle, cette règle sera immédiatement suivie d'une fusion qui fera décroitre $\phi_{3,2}$. S'il n'y avait pas terminaison, on pourrait ainsi extraire une sous-suite infinie sur laquelle Φ_2 serait strictement décroissante, ce qui est absurde.

□

3.7 Validité dans $T(F)$ d'une formule équationnelle

Nous abordons ici la transformation des formules équationnelles. Rappelons que ces formules ne sont que des problèmes équationnels entourés d'un certain nombre de quantificateurs. Comme corollaire aux résultats des sections précédentes, nous pouvons énoncer le:

Théoreme 3.21 *La validité d'une formule équationnelle dans $T(F)$ est décidable.*

On peut aussi énoncer le théorème comme dans [Mal71,Mah88a]:

Il existe une axiomatisation complète des arbres finis sur un alphabet fini de symboles

Ce résultat vient ainsi compléter ceux de [Mal71] sur les classes d'algèbres complètement axiomatisables.

[14] La troisième composante a_3 est la même pour tous les d'_j car nous supposons ici que l'équation décomposée ne contient pas de paramètre

Preuve
Nous appelerons *succession de quantificateurs* une expression $\mathcal{Q}$ définie par la grammaire:

$$\begin{array}{rcl} \mathcal{Q} & \rightarrow & \mathcal{Q}_e \mid \mathcal{Q}_u \mid \epsilon \\ \mathcal{Q}_u & \rightarrow & \forall \vec{z} \mid \forall \vec{z}, \mathcal{Q}_e \\ \mathcal{Q}_e & \rightarrow & \exists \vec{z} \mid \exists \vec{z}, \mathcal{Q}_u \end{array}$$

où "$\forall$", "$\exists$" et "," sont des terminaux et $\vec{z}$ engendre l'ensemble des ensembles finis non vides de variables. La *longueur* d'une succession de quantificateurs est la longueur de la chaine de dérivation permettant de l'engendrer moins 1.

Une formule équationnelle en forme prénexe s'écrit ainsi $\phi \equiv \mathcal{Q} : P$ où $\mathcal{Q}$ est une succession de quantificateurs et P une matrice. Montrons le théorème par récurrence sur la longueur de $\mathcal{Q}$.

- Si $\mathcal{Q}$ est de longueur inférieure ou égale à 2, le théorème 3.20 et la proposition 3.13 montrent le résultat voulu.
- Si $\mathcal{Q}$ est de longueur $n \geq 2$, alors ϕ peut s'écrire -ou bien $\mathcal{Q}', \mathcal{P}$ -ou bien $\mathcal{Q}', \neg\mathcal{P}$, où $\mathcal{P}$ est un problème équationnel et $\mathcal{Q}'$ est une succession de quantificateurs de longueur $n-2$. Dans les deux cas, $\mathcal{P}$ est équivalent (d'après le théorème 3.16) à un ensemble de problèmes $\{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ qui ne contiennent pas de paramètre. Ce qui peut s'écrire $\phi \sim \mathcal{Q}', (\mathcal{P}_1 \vee \ldots \vee \mathcal{P}_n)$ ou bien $\phi \sim \mathcal{Q}', \neg(\mathcal{P}_1 \vee \ldots \vee \mathcal{P}_n)$. Dans les deux cas, après mise en forme prénexe, on obtient une formule équationnelle $\mathcal{Q}'' : P$ où $\mathcal{Q}''$ est de longueur $n-1$. (Le quantificateur universel le plus interne a été "éliminé" par la transformation de la section 3.4). Ce qui achève la preuve, par hypothèse de récurrence.

□

3.8 Complexité de la disunification

Dans les sections qui précèdent nous avons vu des résultats de décidabilité ainsi qu'une façon de construire des algorithmes. Une question naturelle est de savoir si de tels algorithmes sont, en pratique, utilisables. Notons tout d'abord que:

Proposition 3.22 *La décision de l'existence d'une solution dans $T(F)$ à un problème équationnel est un problème NP-dur.*

Preuve
On code aisément le problème de la satisfiabilité en calcul propositionnel, chaque littéral positif x étant transformé en $x = vrai$ et chaque littéral négatif $\neg x$ en $x = faux$. F est alors constitué des deux constantes vrai et faux. Notons que le problème obtenu ne contient même pas de quantificateur.□

La décision de la validité d'une formule équationnelle serait un problème NP-complet ([Mah88b]).

Tout cela ne donne pas vraiment d'indication sur la possibilité d'utiliser effectivement les méthodes des sections précédentes. Il semblerait en fait que la complexité de la disunification vienne de la présence "d'équations dans les disjonctions". Dans le cas du problème de la satisfiabilté en calcul propositionnel, on est contraint d'utiliser la règle de choix non déterministe qui conduit à des duplications de certaines parties du problème. De même si certaines équations ne sont pas résolues, la règle de décomposition (D_3) entraine aussi certaines duplications. À l'inverse, un problème constitué uniquement de conjonctions d'équations (sans quantificateurs) peut être résolu en temps linéaire [PW78]. De même, une disjonction de diséquations, négation du problème précédent, n'est pas un problème complexe. La combinaison des deux cas, comme l'introduction de quantificateurs ne semble pas augmenter de façon appréciable la complexité. En conclusion, bien qu'il n'y ait pas (pour l'instant) pas de fondement rigoureux, il semblerait donc que les problèmes ne comportant pas d'équations dans les disjonctions[15] soient "peu complexes". (Polynomiaux ?)

En vue d'une implantation efficace, il faut de toutes façons

1. Donner un contrôle précis bien choisi.
2. Modifier les règles de transformation en les combinant de façon cohérente avec le contrôle. Par exemple, l'explosion étant toujours suivie d'une fusion, puis d'une décomposition ou d'une incompatibilité, il est judicieux de donner une seule règle qui combine ces trois étapes afin d'éviter de "faire grossir" le problème pour le réduire immédiatement après.

Une étude détaillée de l'implantation de tels algorithmes reste à faire.

[15] Les problèmes que nous utiliserons dans les applications satisfont cett propriété. D'où l'importance de ces considérations

Chapitre 4

Autres formes résolues

Dans ce chapitre nous nous intéressons d'une part à d'autres formes résolues (plus "raffinées" que dans le chapitre précédent), d'autre part à la résolution des problèmes équationnels dans d'autres algèbres que $T(F)$.

Tout d'abord, nous nous intéressons à l'élimination de la négation (chaque fois que c'est possible). Cela permet, par exemple, de représenter explicitement des termes définis par contre-exemples ([LM87]). Cela permet aussi de nombreuses simplifications dans les problèmes liés à la complétude des définitions et aux preuves par induction (cf chapitres 5 et 6).

Dans cette première partie nous donnerons ainsi un contrôle permettant de transformer un problème en un ensemble de problèmes sans diséquations, chaque fois que de telles formes résolues existent.

Nous nous intéressons ensuite à la résolution des problèmes équationnels dans les arbres rationnels. Le chapitre précédent ne donne en effet de résultat que dans le cas des arbres finis. Or on sait bien qu'en programmation (logique) les arbres rationnels jouent un rôle important. Au prix de quelques règles supplémentaires et de résultats techniques, nous montrerons donc comment étendre les résultats du chapitre précédent pour traiter ce cas.

La section 4.3 sera quant à elle consacrée à la résolution des problèmes équationnels dans $T(F, X)$, ou, si l'on préfère, lorsque F est infini. Ce cas, quoique plus simple que celui de $T(F)$, n'est pas une conséquence directe des résultats du chapitre précédent parce que l'on ne peut plus utiliser la règle d'explosion (Ex_1).

La section 4.4 est consacrée à la résolution des problèmes équationnels dans les algèbres avec sortes ordonnées (cf entre autres [GM87b,GKK88,Kir88,SNGM87,Sch87a]). Jusqu'à présent en effet nous n'avons considéré que des algèbres "multi-sortes" mais nous verrons qu'il est possible d'étendre les résultats au cas des sortes ordonnées à peu de frais.

Ensuite, nous montrerons que l'on peut encore "simplifier" les formes résolues du chapitre précédent, par exemple, en éliminant les cycles dans les diséquations. Pour cela, nous introduisons une autre forme de problème équationnel, permettant d'exprimer plus

facilement un certain contrôle (que nous appelons *résolution progressive*).

Enfin, nous étudierons des problèmes particuliers appelés *problèmes de complément* et qui possèdent de nombreuses applications. En particulier, les formes résolues présentées dans cette section seront utilisées dans le chapitre suivant.

4.1 Élimination des diséquations

Nous supposerons dans toute cette section que $\mathcal{A} = T(F)$[1]. Nous supposerons aussi, dans un souci de simplicité, que toute sorte est infinitaire[2].

Nous voulons obtenir ici la propriété supplémentaire suivante sur les formes résolues: si $\mathcal{P}$ est équivalent à un problème d'unification, alors ses formes résolues sont des problèmes d'unification. Ce qui revient à dire qu'on élimine la négation (c'est-à-dire les diséquations) chaque fois que c'est possible. Ce résultat peut être comparé à ceux de Lassez et Marriott [LM87] qui prouvent que certains problèmes (qui sont des cas particuliers de problèmes équationnels) ne peuvent être transformés en des problèmes qui ne comportent que des équations.

Les systèmes de règles de transformation du chapitre 3 peuvent se révéler insuffisants (c'est-à-dire incomplets) dans (au moins) trois cas de figure:

1. La règle de choix non déterministe a été appliquée "trop tôt": $x \neq y \vee x = y$ est transformée par (Nc) en $x \neq y$ d'une part et $x = y$ d'autre part. Alors que, utilisant une règle de fusion, on obtient $\top$.

2. Une diséquation entre une variable et un terme fermé peut être éliminée en employant l'explosion: supposant que $F = \{0 :\rightarrow \underline{s};\ s : \underline{s} \rightarrow \underline{s}\}$, $x \neq 0$ peut être transformé en $\exists w, x = s(w)$.

3. Une disjonction d'équations se réduit à $\top$, éliminant ainsi une diséquation. Par exemple, supposons que F est composée de deux opérateurs: la constante 0 et l'opérateur unaire f. Alors le problème

$$\exists w : x_1 = 0 \vee x_1 = f(w) \vee x_2 \neq x_3$$

est équivalent à $\top$ (lorsque $\mathcal{A} = T(F)$) puisque toute $T(F)$-substitution affecte à x_1 soit 0 soit un terme de la forme $f(w)$. La diséquation $x_2 \neq x_3$ a ainsi été éliminée par "effet de bord".

Le premier cas de figure peut être évité en repoussant "aussi longtemps que possible" la règle de choix non déterministe. Le deuxième cas de figure peut lui aussi être évité en

[1] D'autres exemples d'algèbres seront évoqués dans les sections suivantes. C'est le cas $\mathcal{A} = T(F)$ qui est le plus difficile et qui possède de nombreuses applications, comme nous le verrons là aussi dans les sections et chapitres suivants.

[2] Envisager le cas de sortes finitaires n'est pas plus difficile d'un point de vue théorique : on peut éliminer les variables de sorte finitaire en utilisant la règle (Ex_2). Nous effectuons cette hypothèse pour ne pas compliquer l'exposé.

autorisant l'explosion de x lorsque x est membre d'une diséquation $x \neq t$ où t est un terme fermé. Le troisième exemple ne peut être évité avec les règles dont nous disposons. Nous discuterons donc plus loin les moyens de l'éviter. Mais nous nous limiterons d'abord à des problèmes *ne contenant pas d'équations dans les disjonctions.*

Définition 4.1 *Un problème équationnel est dit* SED *(Sans équations dans les disjonctions) si sa forme normale conjonctive est de la forme* $\exists \vec{w}, \forall \vec{y} : d_1 \wedge \ldots \wedge d_n$ *où chaque* d_i *est ou bien une équation, ou bien une diséquation ou bien une disjonction de diséquations.*

Définition 4.2 *Un problème équationnel* a la propriété EU *s'il est équivalent (par rapport à* $T(F)$, $\mathcal{I}$*) à un problème d'unification.*

Nous allons donc voir que les règles du chapitre précédent permettent de transformer tout problème SED ayant la propriété EU en un ensemble fini de problèmes d'unification. Pour cela, il nous faut tout d'abord caractériser (syntaxiquement) certains problèmes qui n'ont pas la propriété EU. Cela nous permettra de prouver qu'un problème irréductible qui n'est pas un problème d'unification n'est pas équivalent à un problème d'unification.

4.1.1 Résultats de non-équivalence avec les problèmes d'unification

Notons tout d'abord une forme équivalente de la propriété EU:

Lemme 4.3 *Un problème équationnel* $\mathcal{P}$ *a la propriété EU (par rapport à* $\mathcal{I}$*) si et seulement s'il existe un nombre fini (éventuellement nul) de substitutions idempotentes* $\sigma_1, \ldots, \sigma_n$ *de domaine* $\mathcal{I}$ *telles que*

$$\mathcal{S}(T(F), \mathcal{P}, \mathcal{I}) = \bigcup_{i=1,\ldots,n} \{\sigma_i \sigma | \sigma \text{ est une } T(F)\text{-substitution de domaine } VIm(\sigma_i)\}$$

Preuve

Remarquons que les formes irréductibles (pour $\mathcal{R}_2$) de problèmes d'unification sont de la forme

$$\exists \vec{w} : z_1 = t_1 \wedge \ldots \wedge z_n = t_n$$

De plus, un tel problème peut toujours être transformé (par exemple en utilisant les règles (MF_2) et (MF_3)) en un problème de la même forme où, pour tout i, $z_i \in \mathcal{I}$ et $Var(t_i) \cap \mathcal{I} = \emptyset$.

Un problème $\mathcal{P}$ ayant la propriété EU est ainsi équivalent à un ensemble fini de problèmes $\mathcal{P}_1, \ldots, \mathcal{P}_n$ de la forme

$$\mathcal{P}_i \equiv \exists \vec{w_i} : x_1 = t_{1,i} \wedge \ldots \wedge x_n = t_{n,i}$$

où $\{x_1, \ldots, x_n\} = \mathcal{I}$ et $\vec{w_i} = Var(t_{1,i}, \ldots, t_{n,i})$. On note alors σ_i la substitution $\{x_1 \rightarrow t_{1,i}; \ \ldots \ x_n \rightarrow t_{n,i}\}$. Le résultat du lemme est maintenant une conséquence de la définition d'une solution:

$$\begin{aligned} \mathcal{S}(T(F), \mathcal{P}, \mathcal{I}) &= \bigcup_{i=1,\ldots,n} \mathcal{S}(T(F), \mathcal{P}, \mathcal{I}) \\ &= \bigcup_{i=1,\ldots,n} \{\sigma_i \rho | Dom(\rho) = \vec{w_i}\} \end{aligned}$$

□

Le lemme technique suivant, en montrant la construction de deux instances non unifiables de termes dont l'un est une instance de l'autre, nous permettra de prouver que $t \neq u$ ne peut avoir la propriété EU lorsque u est une instance de t. Ce résultat sera progressivement étendu à des disjonctions de diséquations puis à certains problèmes équationnels.

Une construction analogue est d'ailleurs donnée dans [LM87] dans un contexte un peu différent.

Lemme 4.4 *Supposons que F contient un opérateur d'arité au moins égale à 2 ou bien au moins deux opérateurs d'arité 1. Soient t et u deux termes sans variable commune, non fermés et tels que u est une instance de t. Soient $t_1, \ldots, t_n, u_1, \ldots, u_n$ $2n$ termes tels que :*

- *Pour tout i, t_i et u_i ne sont pas unifiables,*
- *$t_1, \ldots, t_n$ sont des instances de t,*
- *$u_1, \ldots, u_n$ sont des instances de u.*

Alors, il existe deux termes t' et u' non fermés et tels que:

- *Il existe une substitution σ telle que $t' \equiv t\sigma$ et $u' \equiv u\sigma$,*
- *pour tout indice i, $Var(t', u') \cap Var(t_i, u_i) = \emptyset$,*
- *pour toute substitution θ et tout indice i, $t_i\theta \equiv t' \Rightarrow u_i\theta \not\equiv u'$,*
- *σ est $Var(t, u)$-linéaire et $\sigma(X) \cap T(F) = \emptyset$.*

Preuve

Soit $N = 1 + \max_{1 \leq i \leq n}(\max(\text{profondeur}(t_i), \text{profondeur}(u_i)))$. Soit f un opérateur d'arité λ maximale.

Soit α un terme de profondeur N tel que:

1. toute suite de N entiers inférieurs ou égaux à λ est une position de α,
2. pour toute position p de longueur strictement inférieure à N, $\alpha_N(p) = f$,
3. α n'a pas de variable commune avec t, u ni avec les termes t_i, u_i.

Autrement dit, comme l'illustre la figure 4.1, α est un arbre ne comportant que des noeuds étiquetés par f et de profondeur N. α_N possède ainsi λ^N variables distinctes.

Soit $M = |Var(u)|$. $\alpha_N, \ldots, \alpha_{N+M-1}$ sont alors M termes obtenus par renommage des variables de α, de façon à ce que deux α_i distincts n'aient pas de variable commune et pas de variable commune avec t_j, u_j.

Soient $x_1, \ldots, x_M$ les variables de u. Soit alors v_1 le terme $u\{x_1 \rightarrow \alpha_N; \ \ldots \ ; x_M \rightarrow \alpha_{N+M-1}\}$ (voir figure 4.2).

Comme F contient un opérateur d'arité supérieure ou égale à 2 ou bien au moins deux opérateurs unaires distincts, il existe dans $T(F, X)$ deux termes v_2 et v_3 non fermés, non unifiables, sans variables communes et sans variable commune avec v_1 ou l'un des t_i, u_i. (Prendre $v_2 \equiv f(a, x_2, \ldots, x_n)$ et $v_2 \equiv f(f(x'_1, \ldots, x'_n), a, \ldots, a)$ dans le cas où f est d'arité

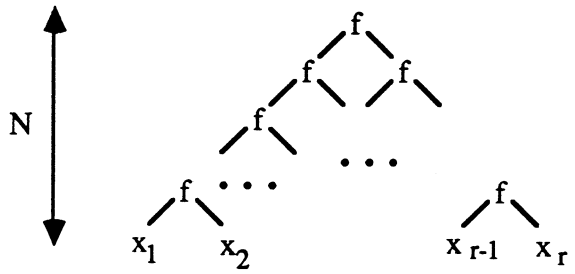

Figure 4.1: construction de α_m

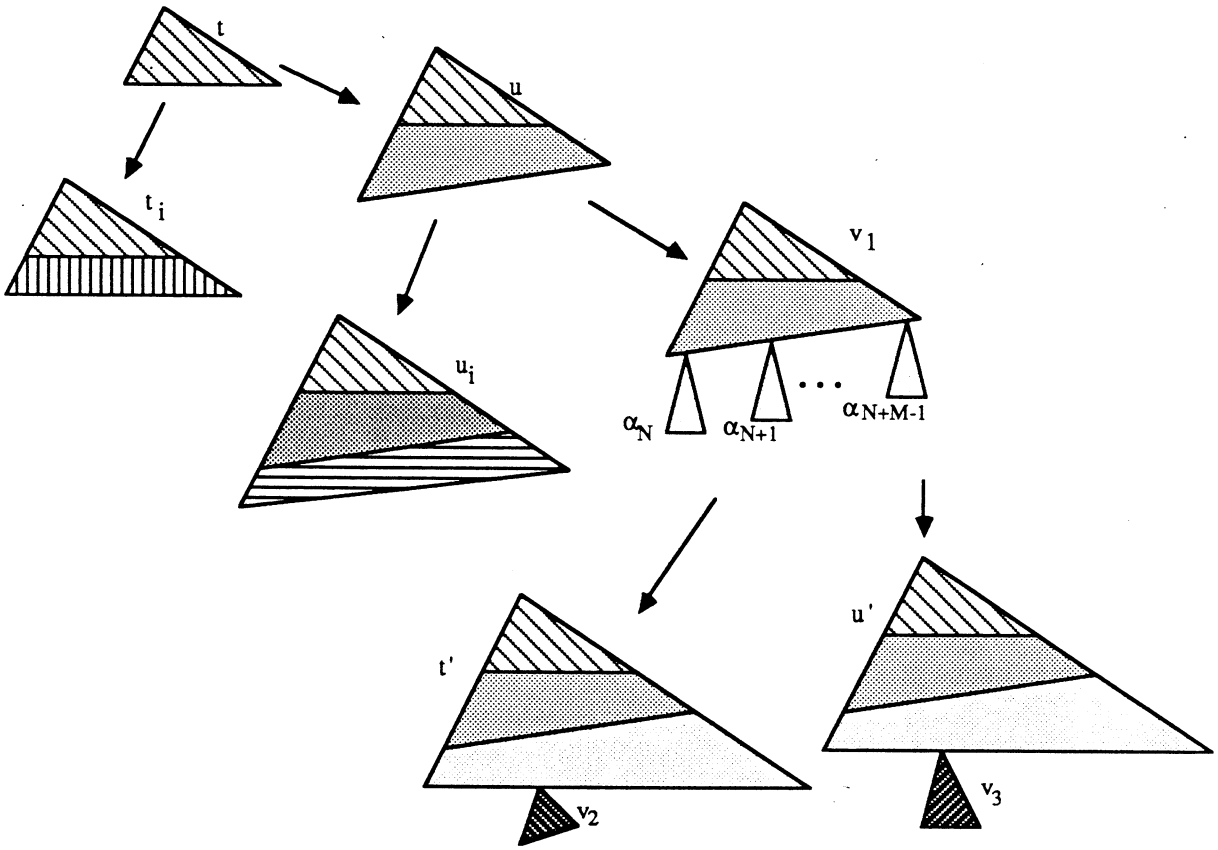

Figure 4.2: Construction de t' et u'

supérieure à 2. Prendre $v_2 \equiv f(x)$ et $v_3 \equiv g(x')$ si f et g sont deux opérateurs unaires distincts).

On pose alors $t' \equiv v_1\{z \rightarrow v_2\}$ et $u' \equiv v_1\{z \rightarrow v_3\}\theta$, où z est une variable de v_1 et θ est un renommage de variables de telle façon que t' et u' n'aient pas de variable commune.

t' et u' n'ont bien sûr pas de variable commune, ne sont pas fermés et ne sont pas unifiables. De plus, comme t et u n'ont pas de variable commune, il existe une substitution σ telle que $t' \equiv t\sigma$ et $u' \equiv u\sigma$. (Par construction). De même, les conditions $\sigma(X) \cap T(F) = \emptyset$ et σ est $Var(t,u)$-linéaire sont satisfaites par construction. Il ne reste donc plus qu'à vérifier la condition 4.

Raisonnons par l'absurde et supposons qu'il existe une substitution θ_0 et un indice i tels que $t_i\theta_0 \equiv t'$ et $u_i\theta_0 \equiv u'$. Nous allons montrer que la substitution σ_0 qui associe à chaque variable $x \in Var(t_i, u_i)$ le terme v_1/p où p est une position de x dans t_i ou dans u_i est bien définie (indépendamment de p) et est un unificateur de t_i et u_i. Ce qui constituera une contradiction.

On peut tout d'abord remarquer que $Pos(t_i) \cup Pos(u_i) \subseteq Pos(v_1)$ car

$$\{p \in Pos(t') \mid |p| \leq N\} = \{p \in Pos(u') \mid |p| \leq N\}$$

Par conséquent, si $x \in Var(t_i) \cup Var(u_i)$ n'a qu'une occurrence dans t_i, u_i, $x\sigma_0 = v_1/p$ définit bien (de manière unique) la valeur de σ_0 en x. Considérons maintenant les cas d'occurrences multiples:

- Si p_1, p_2 sont deux positions de x dans t_i, ce sont des positions de t' et de v_1 de longueur inférieure à N telles que $t'/p_1 \equiv t'/p_2$. Ce qui peut s'écrire

  $$v_1\{z \rightarrow v_2\}/p_1 \equiv v_1\{z \rightarrow v_2\}/p_2.$$

  D'où

  $$(v_1/p_1)\{z \rightarrow v_2\} \equiv (v_1/p_2)\{z \rightarrow v_2\}.$$

  On peut en déduire que $v_1/p_1 \equiv v_1/p_2$ car, s'il existait une position q telle que $v_1/p_1(q) \neq v_1/p_2(q)$,

  - ou bien $v_1/p_1(q)$ et $v_1/p_2(q)$ sont des symboles distincts de z et dans ce cas la différence subsiste après application de $\{z \rightarrow v_2\}$.
  - ou bien $z \in \{v_1/p_1(q), v_1/p_2(q)\}$. Par raison de symétrie, on peut supposer sans perdre de généralité que $z = v_1/p_1(q)$. A nouveau, deux cas se présentent:
    * ou bien $z \in Var(v_1/p_2 \cdot q)$. Dans ce cas v_1/p_1 et v_1/p_2 ne sont pas unifiables.
    * ou bien $z \notin Var(v_1/p_2 \cdot q)$. Dans ce cas, comme, par construction, v_2 contient une variable z_0 qui n'est pas dans v_1, $(v_1/p_1 \cdot q)\{z \rightarrow v_2\}$ contient une occurrence de z_0 alors que $(v_1/p_2 \cdot q)\{z \rightarrow v_2\}$ n'en contient pas.

  et, dans tous les cas, $(v_1/p_1)\{z \rightarrow v_2\}$ serait distinct de $(v_1/p_2)\{z \rightarrow v_2\}$.

- Si p_1 est une position de x dans t_i et p_2 est une position de x dans u_i, $x\theta_0$ ne contient pas d'occurrence de v_2 ni de v_3 (puisque c'est un sous-terme d'à la fois t' et u'). Il en résulte que $t'/p_1 \equiv v_1/p_1 \equiv v_1/p_2 \equiv u'\theta^{-1}/p_2$.

Ainsi, dans tous les cas, si x a une occurrence dans t_i ou u_i, le terme v_1/p est indépendant de l'occurrence p de x choisie et σ_0 est bien définie. De plus, $t_i\sigma_0 \equiv v_1 \equiv u_i\sigma_0$ ce qui conduit à la contradiction. □

Nous pouvons déduire de ce lemme un premier résultat de non équivalence avec un problème d'unification:

Lemme 4.5 *Soient t et u deux termes distincts non fermés tels que u est une instance de t et que $Var(t) \cap Var(u) = \emptyset$. Alors $t \neq u$ n'a pas la propriété EU (par rapport à $\mathcal{I} = Var(t, u)$).*

Preuve

Notons tout d'abord que d'après le lemme 3.7, $t \neq u$ a au moins une solution fermée puisque t et u sont distincts. Raisonnons par l'absurde et supposons que $t \neq u$ a la propriété EU. D'après le lemme 4.3, il existe un ensemble fini de substitutions $\sigma_1, \ldots, \sigma_n$ telles que $u\sigma_i$ et $t\sigma_i$ sont non unifiables et que toute solution de $t \neq u$ est une instance de l'un des σ_i.

Deux cas se présentent alors:

1. F ne contient que des constantes et un symbole unaire g.
   Alors, les hypothèses du lemme entrainent que t est de la forme $g^{m_1}(x)$ et u de la forme $g^{m_2}(x')$ avec $m_1 \leq m_2$. $t \neq u$ est alors équivalent à $x \neq g^{m_2-m_1}(x')$. Par hypothèse, $x\sigma_i$ et $g^{m_2-m_1}(x'\sigma_i)$ ne sont ainsi pas unifiables. Il en résulte que $x\sigma_i \equiv g^{k_i}(a)$ et $g^{m_2-m_1}(x'\sigma_i) \equiv g^{r_i}(b)$ avec

   - ou bien $k_i \neq r_i$ et au moins l'un des deux termes a et b est une constante
   - ou bien a et b sont des constantes distinctes

   Dans un cas comme dans l'autre, posons $K = \max_{i=1,\ldots,n}(\max(k_i, r_i))$. La substitution
   $$\sigma_0 = \{x \rightarrow g^{K+1+m_2-m_1}(a);\ x' \rightarrow g^{K+2}(a)\}$$
   où a est une constante quelconque, est solution de $t \neq u$ ($t\sigma_0 \equiv g^{K+1+m_2}(a) \not\equiv g^{K+2+m_2}(a) \equiv u\sigma_0$). σ_0 n'est instance d'aucune des substitutions σ_i, en effet, dans tous les cas, au moins l'un des deux termes $u\sigma_i$ ou $t\sigma_i$ est un terme fermé de la forme $g^{k+m_2}(b)$ où $k \leq K$ et b est une constante.

   On obtient donc une contradiction.

2. F contient au moins un symbole d'arité supérieure ou égale à 2 ou bien deux opérateurs unaires.
   Si l'on pose alors $t_i \equiv t\sigma_i$ et $u_i \equiv u\sigma_i$, t, u, t_i et u_i vérifient les hypothèses du lemme 4.4. Par conséquent, il existe une substitution σ_0 telle que $u\sigma_0$ et $t\sigma_0$ ne sont pas unifiables et telle que, pour tout indice i, -ou bien $u\sigma_0$ n'est pas une instance de $u\sigma_i$ -ou bien $t\sigma_0$ n'est pas une instance de $t\sigma_i$. Si ρ est une substitution fermée quelconque de domaine $VIm(\sigma_0)$, $\rho \circ \sigma_0$ est une solution de $t \neq u$ et n'est une instance d'aucun des σ_i. Ce qui donne la contradiction voulue.□

Généralisons maintenant ce résultat aux systèmes de diséquations:

Lemme 4.6 *Soit* $\mathcal{P} \equiv u_1 \neq t_1 \wedge \ldots \wedge u_n \neq t_n$ *un problème équationnel tel que*

- *pour tout indice* i, t_i *et* u_i *sont des termes non fermés tels que* $Var(t_i) \cap Var(u_i) = \emptyset$
- *pour tout* i, u_i *est une instance de* t_i.

Alors $\mathcal{P}$ *n'a pas la propriété EU.*

Preuve

Si F ne contient qu'un opérateur unaire et des constantes, un raisonnement analogue à celui que nous avons effectué ci-dessus conduit au résultat. Nous supposerons donc dans la suite de cette preuve que F contient au moins un opérateur d'arité supérieure ou égale à 2 ou bien deux opérateurs unaires distincts.

Nous allons prouver par récurrence sur n (c'est-à-dire le nombre de diséquations) que, si $\sigma_1, \ldots, \sigma_m$ sont m substitutions telles que, pour tous indices i et j $u_i\sigma_j$ et $t_i\sigma_j$ ne sont pas unifiables, alors il existe une substitution σ_0 telle que:

1. pour tout indice i, $u_i\sigma_0$ et $t_i\sigma_0$ ne sont pas unifiables
2. $\sigma(X) \cap T(F) = \emptyset$ et σ_0 est $Var(\mathcal{P})$-linéaire
3. σ_0 n'est instance d'aucune des substitutions σ_j

Le lemme 4.6 sera alors une conséquence immédiate du lemme 4.3.

Lorsque $n = 1$, ce résultat est une conséquence du lemme 4.4, comme il a été montré dans la preuve du lemme 4.5.

Supposons maintenant la propriété vraie pour $n - 1$. Supposant que toute instance de l'une des substitutions $\sigma_1, \ldots, \sigma_m$ est solution de $\mathcal{P}$ (c'est-à-dire que pour tous i, j, $t_i\sigma_j$ et $u_i\sigma_j$ ne sont pas unifiables), par hypothèse de récurrence, il existe une substitution σ_{m+1} qui n'est comparable à aucune des σ_i, $i = 1, \ldots, m$ et telle que toutes ses instances sont solutions de

$$u_1 \neq t_1 \wedge \ldots \wedge u_{n-1} \neq t_{n-1}$$

σ_{m+1} est de plus $Var(t_1, u_1, \ldots, t_n, u_n)$-linéaire et l'image d'une variable quelconque n'est pas un terme fermé. On peut même supposer aussi que σ_{m+1} est idempotente. Les termes $t_n\sigma_{m+1}$ et $u_n\sigma_{m+1}$ sont sans variable commune et ne sont pas fermés, par propriété de σ_{m+1}. Si ces deux termes ne sont pas unifiables, alors il suffit de choisir $\sigma_0 = \sigma_{m+1}$. Dans le cas contraire, soit θ un plus général unificateur idempotent de $t_n\sigma_{m+1}$ et $u_n\sigma_{m+1}$. Quatre cas se présentent alors:

1. Il existe une variable z telle que $z\theta$ est non variable et, de plus, F possède au moins deux opérateurs non constants.
   Soit v un terme linéaire non fermé, sans variable dans $VIm(\sigma_{m+1})$ et non unifiable avec $z\theta$. Un tel terme existe : il suffit de choisir pour v un terme dont la racine est distincte de celle de $z\theta$. Il suffit alors de poser $\sigma_0 = \{z \rightarrow v\}\sigma_{m+1}$.

2. $\theta(X) \subseteq X$.
   θ étant un plus général unificateur de $t_n\sigma_{m+1}$ et de $u_n\sigma_{m+1}$, cette condition signifie en particulier que $Pos(t_n\sigma_{m+1}) = Pos(u_n\sigma_{m+1}) = POS$. Soit alors $p \in POS$ telle que $t_n\sigma_{m+1}/p \equiv z$ est une variable. Alors $u_n\sigma_{m+1}/p \equiv z'$ est aussi une variable

(distincte de z par hypothèse). Soient v_1 et v_2 deux termes linéaires non fermés, non unifiables et sans variables communes tels que $VIm(\sigma_{m+1}) \cap Var(v_1, v_2) = \emptyset$. (Deux tels termes existent comme vu dans la preuve du lemme 4.4). Il suffit alors de poser $\sigma_0 = \{z \rightarrow v_1;\ z' \rightarrow v_2\}\sigma_{m+1}$.

3. F contient comme seul symbole de fonction non constant le symbole f d'arité supérieure ou égale à 2 et il existe une variable z telle que $z\theta$ ne soit ni une variable ni de la forme $f(z_1, \ldots, z_r)$ où $z_1, \ldots, z_r$ sont des variables.
Alors deux cas se présentent à nouveau:

   (a) $z\theta$ contient une occurrence de constante.
   Soit $y\theta/p \equiv a$ une constante. Soit alors $v_1 \equiv y\theta[f(z_1, \ldots, z_r)]_p$ où $z_1, \ldots, z_r$ sont des variables distinctes n'ayant pas d'autres occurrences. Soit ρ la substitution qui associe a à toute variable de $y\theta$. Il suffit alors de poser $\sigma_0 = \{z \rightarrow v_1\rho\}\sigma_{m+1}$.

   (b) $z\theta$ ne contient pas d'occurrence de constante.
   Soit $z\theta \equiv f(v_1, \ldots, v_r)$ et v_i un terme non variable. On pose alors $v_0 \equiv f(v_1, \ldots, v_{i-1}, a, v_{i+1}, \ldots, v_r)\rho$ où a est une constante et ρ associe à toutes les variables de $Var(v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_r)$ sauf une la constante a. (Rappelons que $r \geq 2$ et qu'il n'y a pas de constante dans $z\theta$. Une telle construction est donc possible). Il suffit alors de choisir $\sigma_0 = \{z \rightarrow v_0\}\sigma_{m+1}$.

4. F contient comme seul symbole de fonction non constant le symbole f d'arité supérieure ou égale à 2 et, pour toute variable z, $z\theta$ est ou bien une variable ou bien de la forme $f(z_1, \ldots, z_r)$.
On peut supposer que nous ne sommes pas dans le deuxième cas ci-dessus, c'est-à-dire que, pour au moins une variable z, $z\theta$ n'est pas variable. Soit alors p une position telle que $t_n\sigma_{m+1}/p \equiv z$ et $z\theta \equiv f(z_1, \ldots, z_r)$. (Le cas où p est une position de $u_n\sigma_{m+1}$ est identique). Par minimalité de θ, p est alors une position de $u_n\sigma_{m+1}$ et $u_n\sigma_{m+1}/p$ est ou bien une variable (qui a au moins deux occurrences) ou bien de la forme $f(z'_1, \ldots, z'_r)$. A nouveau, construisons σ_0 dans chacun de ces cas:

   - $u_n\sigma_{m+1}/p \equiv z' \in X$.
   Soient à nouveau v_1 et v_2 deux termes linéaires non fermés et non unifiables. On pose
   $$\sigma_0 = \{z \rightarrow f(v_1, z_2, \ldots, z_r);\ z' \rightarrow f(v_2, z''_2, \ldots, z''_r)\}\sigma_{m+1}$$
   - $u_n\sigma_{m+1}/p \equiv f(z'_1, \ldots, z'_r)$.
   On pose
   $$\sigma_0 = \{z \rightarrow f(v_1, z_2, \ldots, z_r);\ z' \rightarrow v_2\}\sigma_{m+1}$$

□

Généralisons maintenant le résultat précédent à des conjonctions de disjonctions de diséquations.

Lemme 4.7 *Soit $\mathcal{P} \equiv d_1 \wedge \ldots \wedge d_n$ où chaque d_i est de la forme*

$$z_{1,i} \neq t_{1,i} \vee \ldots \vee z_{m,i} \neq t_{m,i}$$

$z_{1,i}, \ldots, z_{m,i}$ étant des variables distinctes et n'ayant pas d'autre occurrence dans d_i. On suppose de plus que, pour tout i, il existe au moins un j tel que $t_{i,j} \notin T(F)$. Alors $\mathcal{P}$ n'a pas la propriété EU.

Preuve

Pour chaque d_i, on ajoute à F un opérateur f_i d'arité égale au nombre de diséquations de d_i. On obtient ainsi une signature F'. Le problème $\mathcal{P}'$ obtenu en remplaçant d_i par $f_i(z_{1,i}, \ldots, z_{m,i}) \neq f(t_{1,i}, \ldots, t_{m,i})$ vérifie alors les hypothèses du lemme 4.6 et n'a donc pas la propriété EU (par rapport à F'). Comme $\mathcal{P}$ et $\mathcal{P}'$ sont équivalents par rapport à F', il en résulte que $\mathcal{P}$ n'a pas la propriété EU, par rapport à F'. Mais notons que, si $\mathcal{P}$ est équivalent à un problème d'unification par rapport à $T(F)$, alors il est équivalent à un problème d'unification par rapport à $T(F')$. (L'implication contraire étant fausse). D'où le résultat. □

Généralisons maintenent en ajoutant la possibilité d'occurrences d'équations dans les problèmes:

Lemme 4.8 *Soit $\mathcal{P} \equiv \exists \vec{w} : P$ un problème équationnel sans paramètre. Soit x une variable de $\mathcal{I}$ n'ayant pas d'occurrence dans $\mathcal{P}$ et t un terme tel que $Var(t) \cap \mathcal{I} = \emptyset$. $\exists(\vec{w} - Var(t)) : P$ est alors équivalent (par rapport à $\mathcal{I} \cup Var(t)$) à un problème d'unification ssi $\mathcal{P} \wedge x = t$ est équivalent (par rapport à $\mathcal{I}$) à un problème d'unification.*

Preuve

⇒ Supposons ici que $\exists(\vec{w} - Var(t)) : P$ est équivalent (par rapport à $\mathcal{I} \cup Var(t)$) à un problème d'unification.
D'après le lemme 4.3, il existe des substitutions idempotentes $\sigma_1, \ldots, \sigma_n$ de domaine $\mathcal{I} \cup Var(t)$ telles que les solutions de $\exists(\vec{w} - Var(t)) : P$ soient les substitutions $\sigma_i\sigma$, pour σ de domaine $VIm(\sigma_i)$. On peut supposer de plus sans perdre de généralité que $\forall i, x\sigma_i \equiv x$ puisque $x \notin Var(P)$. Soient alors $\sigma'_i = (\{x \rightarrow t\}\sigma_i)|_{\mathcal{I}}$. Nous allons prouver que les solutions de $\mathcal{P} \wedge x = t$ sont les substitutions $\sigma'_i\sigma'$ pour $Dom(\sigma') = VIm(\sigma'_i) = VIm(\sigma_i)$. Cela permettra alors d'utiliser le lemme 4.3 pour déduire que $\mathcal{P} \wedge x = t$ a la propriété EU.

Il reste donc deux inclusions à prouver pour avoir l'égalité des ensembles de substitutions:

- Soit σ' une substitution fermée quelconque de domaine $VIm(\sigma'_i)$. Nous voulons montrer ici que $\sigma'_i\sigma'$ est une solution de $\mathcal{P} \wedge x = t$. Il nous faut donc construire une substitution ρ' de domaine $\vec{w}$ telle que $\sigma'_i\sigma'\rho'$ valide $P \wedge x = t$. Notons $\sigma_{i,1}$ la restriction de σ_i à $\mathcal{I}$, $\sigma_{i,2}$ la restriction de σ_i à $Var(t)$ (on a alors $\sigma_i = \sigma_{i,1}\sigma_{i,2}$), σ'_1 la restriction de σ' à $VIm(\sigma_{i,2})$ et σ'' la substitution $\{x \rightarrow t\sigma_i\}\sigma'_1$. Soit tout d'abord ρ'_1 la substitution fermée de domaine $Var(t)$ définie par : $z\rho'_1 \equiv z\sigma_i\sigma'$ pour toute variable $z \in Var(t)$. Alors $\sigma'_i\rho'_1$ a pour domaine $Var(t) \cup \mathcal{I}$ et

$$\begin{aligned} \sigma'_i\rho'_1 &= \{x \rightarrow t\sigma_i\}\sigma_{i,1}\rho'_1 \\ &= \{x \rightarrow t\sigma_i\}\sigma_{i,1}\sigma_{i,2}\sigma'_1 \\ &= \sigma_i\sigma'' \end{aligned}$$

Soit alors ρ'_2 une substitution fermée quelconque de domaine $VIm(\sigma_i\sigma'')$ et soit $\sigma = \sigma''\rho'_2$. On a l'identité:

$$\sigma'_i\rho'_1\rho'_2 = \sigma_i\sigma''\rho'_2 = \sigma_i\sigma$$

Par hypothèse, $\sigma_i\sigma$ est une solution de $\exists(\vec{w} - Var(t)) : P$ et il existe donc une substitution fermée ρ'_3 dont le domaine est $\vec{w} - Var(t)$ et telle que $\sigma'_i\rho'_1\rho'_2\rho'_3$ valide P. Il suffit alors de poser $\rho' = \rho'_1\rho'_2\rho'_3$.

- Inversement, si θ est une solution de $\mathcal{P} \wedge x = t$, alors il existe une substitution fermée ρ' de domaine $\vec{w}$ telle que $\theta\rho'$ valide $P \wedge x = t$. Soient alors ρ_1 la restriction de ρ' à $Var(t)$ et ρ_2 la restriction de ρ' à $\vec{w} - Var(t)$ ($\rho = \rho_1\rho_2$). $\theta\rho_1$ est une solution de $\exists(\vec{w} - Var(t)) : P$ et par conséquent de la forme $\sigma_i\sigma$. Alors, $\theta = (\{x \rightarrow t\}\sigma_i\sigma)|_{\mathcal{I}}$ est bien de la forme $\sigma'_i\sigma'$. Ce qui prouve l'inclusion inverse.

$\Leftarrow$ Supposons que $\mathcal{P} \wedge x = t$ est équivalent à un problème d'unification. De même que ci-dessus, on utilise le lemme 4.3: $\sigma'_1, \ldots, \sigma'_n$ sont des substitutions telles que l'ensemble des solutions de $\mathcal{P} \wedge x = t$ coincide avec l'ensemble des substitutions de la forme $\sigma'_i\sigma'$. L'équation $x\sigma'_i = t$ a alors au moins une solution θ. Par conséquent $x\sigma'_i$ et t (qui n'on pas de variables communes) sont unifiables; soit α un plus général unificateur (idempotent) de ces deux termes. On note alors σ_i la restriction à $X - \{x\}$ de $\sigma'_i\alpha$. Là encore, nous allons prouver que l'ensemble des solutions de $\exists(\vec{w} - Var(t)) : P$ est l'ensemble des substitutions de la forme $\sigma_i\sigma$, ce qui prouvera que ce problème est équivalent à un problème d'unification, par application du lemme 4.3. Deux inclusions sont à prouver:

- Supposons que σ est une substitution quelconque de domaine $VIm(\sigma_i)$. soit γ la restriction à $VIm(\sigma'_i)$ de $\alpha\sigma$. $\sigma'_i\gamma$ est solution de $\mathcal{P} \wedge x = t$ par hypothèse. Il existe donc une substitution fermée ρ' de domaine $\vec{w}$ telle que $\beta = \sigma'_i\gamma\rho'$ valide $P \wedge x = t$. Soit ρ la restriction de ρ' à $\vec{w} - Var(t)$. Montrons que $\sigma_i\sigma\rho$ et la restriction β' de β à $X - \{x\}$ coincident:
  - si $x_0 \in \mathcal{I} - \{x\}$, alors $x_0\beta \equiv x_0\sigma'_i\alpha\sigma\rho' \equiv x_0\sigma_i\sigma\rho$
  - si $z \in Var(t)$, remarquons que

$$\begin{aligned} t\beta &\equiv x\beta \\ &\equiv x\sigma'_i\gamma\rho' \\ &\equiv x\sigma'_i\alpha\sigma\rho' \\ &\equiv t\alpha\sigma\rho' \\ &\equiv t\alpha\sigma \\ &\equiv t\sigma'_i\alpha\sigma \\ &\equiv t\sigma_i\sigma\rho \end{aligned}$$

    d'où $z\beta \equiv z\sigma_i\sigma\rho$.
  - Si $w \in \vec{w} - Var(t)$, alors $w\beta \equiv w\rho' \equiv w\rho \equiv w\sigma_i\sigma\rho$

  On en déduit que $\beta' = \sigma_i\sigma\rho$. Comme β valide $P \wedge x = t$ et que $x \notin Var(P)$, cela signifie que $\sigma_i\sigma\rho$ valide P. $\sigma_i\sigma$ est donc solution de $\exists(\vec{w} - Var(t)) : P$.

- Soit maintenant θ une solution quelconque de $\exists(\vec{w} - Var(t)) : P$. Il nous faut montrer que θ est de la forme $\sigma_i\sigma$. Soit ρ une substitution fermée de domaine $\vec{w} - Var(t)$ telle que $\theta\rho$ valide P. Soit θ_1 la restriction de θ à $Var(t)$ et θ_2 sa restriction à $\mathcal{I}$. ($\theta = \theta_1\theta_2$).
  Alors, $\{x \rightarrow t\theta\rho\}\theta\rho$ valide $x = t \wedge P$. On en déduit que $\{x \rightarrow t\theta\rho\}\theta_2$ est une solution de $\mathcal{P} \wedge x = t$ et s'écrit par conséquent $\{x \rightarrow t\theta\rho\}\theta\rho = \sigma'_i\sigma'$. Mais

$$t\theta_1 \equiv x\{x \rightarrow t\theta\rho\}\theta_2 \equiv x\sigma'_i\sigma'$$

  $\theta_1\sigma'$ est donc un unificateur de $x\sigma'_i$ et de t. Il existe ainsi une substitution δ telle que $\theta_1\sigma' = \alpha\delta$. Notons enfin σ la restriction de δ à $VIm(\sigma_i)$. On a:

$$\{x \rightarrow t\theta\rho\}\theta = \sigma'_i\theta_1\sigma' = \sigma'_i\alpha\delta$$

  et donc $\theta = \sigma_i\sigma$. Ce qui prouve bien l'inclusion inverse.

□

Remarque
Il n'est pas vrai que $\mathcal{P}$ a la propriété EU (par rapport à $\mathcal{I}$) ssi $\mathcal{P} \wedge x = t$ a la propriété EU (par rapport à $\mathcal{I}$). En effet, considérons le problème $\exists w_1, w_2 : x = f(w_1, w_2) \wedge y \neq z$. Il n'est pas équivalent à un problème d'unification (d'après le lemme 4.8) alors que $\exists w_1, w_2 : w_1 \neq w_2$ est équivalent à $\top$. Cela prouve la nécéssité d'enlever les variables de t à $\vec{w}$ dans l'énoncé du lemme 4.8.

Théoreme 4.9 *Soit* $\mathcal{P} \equiv \exists\vec{w} : x_1 = t_1 \wedge \ldots \wedge x_n = t_n \wedge d_1 \wedge \ldots \wedge d_m$ *où*

- $\mathcal{I} = \{x_1, \ldots, x_n\}$ *et* $Var(t_1, \ldots, t_n) \cap \mathcal{I} = \emptyset$
- $Var(d_1, \ldots, d_n) \subseteq Var(t_1, \ldots, t_n)$
- $x_1, \ldots, x_n$ *sont distincts*
- d_i *est de la forme* $z_{1,i} \neq u_{1,i} \vee \ldots \vee z_{k,i} \neq u_{k,i}$ *où les* $z_{j,i}$ *sont des variables n'ayant qu'une occurrence dans* d_i.
- *Pour tout i, il existe dans* d_i *au moins une diséquation dont les membres ne sont ni l'un ni l'autre des termes fermés.*

Alors $\mathcal{P}$ *a la propriété EU si et seulement si* $\mathcal{P}$ *est un problème d'unification.*

Preuve
C'est une conséquence des lemmes 4.7 et 4.8. □

4.1.2 Elimination des diséquations

Les règles données dans les figures 4.3,4.4,4.5 et 4.6 et qui sont des instances des règles données dans le chapitre 3 section 3.1[3] définissent un système que nous noterons $\mathcal{R}_3$. Ce système nous permettra d'éliminer les diséquations chaque fois que c'est possible.

[3]Ces règles ont donc été prouvées correctes et adéquates lorsque $\mathcal{A} = T(F)$ dans le chapitre 3

Pour la circonstance, nous avons modifié la notion de variable presque résolue: $z \in Inc(\mathcal{P}) \cup \mathcal{I}$ est *presque résolue* dans $\mathcal{P}$ si la forme normale conjonctive de $\mathcal{P}$ peut s'écrire $\mathcal{P} \equiv \exists \vec{w}, \forall \vec{y} : z = t \wedge P$ où $t \notin \mathcal{I}$ et $z \notin Var(t)$. Une variable est alors résolue dans $\mathcal{P}$ si elle est presque résolue dans $\mathcal{P}$ et n'a qu'une occurrence dans $\mathcal{P}$. (Cette définition coincide avec la précédente).

Montrons tout d'abord la terminaison:

Théoreme 4.10 *$\mathcal{R}_3$ est à terminaison finie.*

Preuve

La preuve est analogue à celle du théorème 3.20. Il y a en effet peu de changements dans le contrôle. Enonçons quand même la fonction d'interprétation utilisée ainsi que le résumé de ses variations.

$\Phi_3(\exists \vec{w}, \forall \vec{y} : d_1 \wedge \ldots \wedge d_n)$ est le multi-ensemble $\{\psi_3(d_1), \ldots, \psi_3(d_n)\}$ où $\psi_3(d)$ est le 9-uple $(\phi_{3,1}(d), \ldots, \phi_{3,9}(d))$ avec les définitions suivantes:

$\phi_{3,1}(d)$ est le nombre de paramètres distincts ayant une occurrence dans d.

$\phi_{3,2}(e_1 \vee \ldots \vee e_m)$ est le multi-ensemble $\{TM(e_1), \ldots, TM(e_m)\}$ où

- $TM(e) = 0$ si l'un des membres de e est un paramètre résolu
- Sinon, $TM(s = t) = TM(s \neq t) = \max(\text{taille-param}(s), \text{taille-param}(t))$

$\phi_{3,3}(d)$ est le nombre d'équations et de diséquations de d dont un des membres est une variable et l'autre est un terme non variable contenant au moins une occurrence de paramètre.

$\phi_{3,4}(d)$ est le couple (a_1, a_2) formé - du nombre a_1 d'inconnes de $\mathcal{I}$ qui ne sont pas presque résolues dans $\mathcal{P}$ -du nombre a_2 d'inconnues auxiliaires de $\mathcal{P}$ qui ne sont pas presque résolues dans $\mathcal{P}$.

$\phi_{3,5}(d)$ est le nombre d'inconnues de $Inc(\mathcal{P}) \cup \mathcal{I}$ qui ne sont pas résolues dans $\mathcal{P}$.

$\phi_{3,6}(d)$ est le nombre d'inconnues de $\mathcal{P}$ qui ne sont pas localement résolues dans d. Une inconnue z est *localement résolue* dans d (et par rapport à $\mathcal{P}$) s'il existe une diséquation $z \neq t$ de d telle que:

- z n'a qu'une occurrence dans d
- Aucune des inconnues de d n'est presque résolue dans $\mathcal{P}$
- Il n'y a dans $\mathcal{P}$ aucune diséquation $z \neq u$ où $u \in T(F)$

$\phi_{3,7}(e_1 \vee \ldots \vee e_m)$ est le multi-ensemble $\{T'(e_1), \ldots, T'(e_m)\}$ où

- $T'(e) = 0$ si l'un des membres de e est une inconnue résolue
- $T'(z \neq t) = 0$ si les conditions suivantes sont remplies:
- $T'(y \neq u) = 0$ si y est un paramètre
- $T'(z \neq a) = 3$ si z est une inconnue et a une constante

Elimination des équations et diséquations triviales (T)

$$\begin{array}{lll} (T_1) & s = s & \mapsto \top \\ (T_2) & s \neq s & \mapsto \bot \end{array}$$

Fusions (F)

$$\begin{array}{lll} (F_1) & z = t \wedge z = u & \mapsto z = t \wedge t = u \\ (F_3) & z = t \wedge z \neq u & \mapsto z = t \wedge t \neq u \\ (F'_1) & z = t \wedge (z = u \vee d) & \mapsto z = t \wedge (t = u \vee d) \\ (F'_3) & z = t \wedge (z \neq u \vee d) & \mapsto z = t \wedge (t \neq u \vee d) \end{array}$$

Pour ces règles de fusion on supposera que :

1. z est une inconnue et t n'est pas une variable
2. t ne contient pas d'occurrence de paramètre
3. - ou bien u contient une occurrence de paramètre et n'est pas lui-même un paramètre
   - ou bien taille(t) $\leq$ taille(u) et, si u est une inconnue, t n'est pas une constante
   - ou bien u est une variable résolue

$$\begin{array}{lll} (F_2) & z \neq t \vee z \neq u & \mapsto z \neq t \vee t \neq u \\ (F_4) & z = u \vee z \neq t & \mapsto u = t \vee z \neq t \end{array}$$

Pour ces règles de fusion, on supposera que :

1. z est une variable et pas t
2. - ou bien taille-param(t) $\leq$ taille-param(u) et taille-param(u)$\neq 0$
   - ou bien u est un paramètre résolu
   - ou bien taille(t) $\leq$ taille(u), t ne contient pas de paramètre et, lorsque u est une inconnue, t n'est pas une constante.

Décompositions (D)

$$\begin{array}{lll} (D_1) & f(t_1,\ldots,t_n) = f(u_1,\ldots,u_n) & \mapsto t_1 = u_1 \wedge \ldots \wedge t_n = u_n \\ (D_2) & f(t_1,\ldots,t_n) \neq f(u_1,\ldots,u_n) & \mapsto t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \\ (D_3) & f(t_1,\ldots,t_n) = f(u_1,\ldots,u_n) \vee d & \mapsto (t_1 = u_1 \vee d) \wedge \ldots \wedge (t_n = u_n \vee d) \end{array}$$

Figure 4.3: Elimination des diséquations : premier ensemble de règles

Elimination des paramètres (EP)

$$(EP_1) \quad \forall \vec{y}, y : P \quad \mapsto \quad \forall \vec{y} : P$$

si $y \notin Var(P)$.

$$(EP_2) \quad \forall \vec{y} : P \wedge (y \neq t \vee d) \quad \mapsto \quad \forall \vec{y} : P \wedge d\{y \rightarrow t\}$$

Si d est une disjonction d'équations et de diséquations et que $y \in \vec{y}$, $y \notin Var(t)$.

$$(EP_3) \quad \forall \vec{y} : P \wedge (z_1 = u_1 \vee \ldots \vee z_n = u_n \vee R) \quad \mapsto \quad \forall \vec{y} : P \wedge R$$

Si

1. Pour tout i, z_i est une variable et est distincte de u_i
2. Pour tout i, $z_i = u_i$ contient au moins une occurrence de paramètre
3. Pour tout i et tout paramètre $y \in Var(z_i, u_i)$, y est infinitaire
4. R ne contient pas d'occurrence de paramètre

Incompatibilités (I)

$$\begin{array}{lllll} (I_1) & f(t_1,\ldots,t_n) = g(u_1,\ldots,u_m) & \mapsto & \bot & \text{si } f \neq g \\ (I_2) & f(t_1,\ldots,t_n) \neq g(u_1,\ldots,u_m) & \mapsto & \top & \text{si } f \neq g \end{array}$$

Tests d'occurrence (O)

$$\begin{array}{lllll} (O_1) & z = t & \mapsto & \bot & \text{si } z \in Var(t) \text{ et } z \not\equiv t \\ (O_2) & z \neq t & \mapsto & \top & \text{si } z \in Var(t) \text{ et } z \not\equiv t \end{array}$$

Figure 4.4: Elimination des diséquations : deuxième ensemble de règles

Remplacements (R)

$$(R_1) \quad z = t \wedge P \quad \mapsto \quad z = t \wedge P\{z \to t\}$$

Si z est une variable, $z \notin Var(t)$, $z \in Var(P)$, $t \notin \mathcal{I}$, t ne contient pas de paramètre et

- ou bien $t \notin X$
- ou bien t a une occurrence dans P
- ou bien $z \in \mathcal{I}$

$$(R_2) \quad \mathcal{P}[P \wedge (z \neq t \vee Q)] \quad \mapsto \quad \mathcal{P}[P \wedge (z \neq t \vee Q\{z \to t\})]$$

Si

1. z est une inconnue, $z \notin Var(t)$ et $z \in Var(Q)$,
2. t ne contient pas de paramètre,
3. aucune variable de $z \neq t \vee Q$ n'est presque résolue
4. z n'est membre d'aucune diséquation $z \neq u$ dans $\mathcal{P}$ telle que $u \in T(F)$.

Explosion (E)

$$(Ex_1) \quad \forall \vec{y} : P \quad \mapsto \quad \exists w_1, \ldots, w_p, \forall \vec{y} : \; P \wedge x = f(w_1, \ldots, w_p)$$

Cette règle ne sera appliquée que si :

1. x est une inconnue, $\vec{w} \cap (Var(P) \cup \vec{y} \cup \mathcal{I}) = \emptyset$ et $f \in F$
2. Aucune autre règle n'est applicable
3. - ou bien il existe une équation $x = u$ (ou une diséquation $x \neq u$) dans P telle que u n'est pas une variable et u contienne au moins une occurrence de paramètre
   - ou bien il existe une diséquation $x \neq u$ dans P telle que u soit un terme fermé

Figure 4.5: Elimination des diséquations : troisième ensemble de règles

Mise en forme des résultats (MF)

$$(MF_1) \quad \exists w, \mathcal{P} \quad \mapsto \quad \mathcal{P}$$

Si $w \notin Var(\mathcal{P})$

$$(MF_2) \quad \exists \vec{w}, w : w = t \wedge P \quad \mapsto \quad \exists \vec{w} : P$$

Si $w \notin Var(P, t)$, t ne contient pas de paramètre et $t \notin \mathcal{I}$.

$$(MF_3) \quad \exists \vec{w}, \forall \vec{y} : P \quad \mapsto \quad \exists \vec{w}, w, \forall \vec{y} : P \wedge x = w$$

Si $x \in \mathcal{I}$, x n'est pas presque résolue dans $\mathcal{P}$, w est de même sorte que x et $w \notin Var(\vec{w}, \vec{y}, P)$.

$$(MF_4) \quad \exists \vec{w} : (d_1 \vee z_1 \neq u_1) \wedge \ldots \wedge (d_n \vee z_n \neq u_n) \wedge P \quad \mapsto \quad \exists \vec{w} : P$$

Si

1. pour tout i, d_i est une disjonction d'équations et de diséquations
2. pour tout i, z_i est une inconnue
3. pour tout i, $z_i \not\equiv u_i$
4. il existe une variable $w \in \vec{w} \cap Var(z_1, u_1) \cap \ldots \cap Var(z_n, u_n)$ qui n'apparait pas dans P et qui est infinitaire.
5. pour tout i, u_i ne contient pas de paramètre

Figure 4.6: Elimination des diséquations : quatrième ensemble de règles

	$\phi_{3,1}$	$\phi_{3,2}$	$\phi_{3,3}$	$\phi_{3,4}$	$\phi_{3,5}$	$\phi_{3,6}$	$\phi_{3,7}$	$\phi_{3,7}$	$\phi_{3,9}$
$(F_2),(F_4)$	$=$	$=$	$\leq$	$=$	$=$	$=$	$\leq$	$<$	
$(F_1),(F_3),(F_1'),(F_3')$	$=$	$=$	$\leq$	$=$	$\leq$	$=$	$\leq$	$<$	
$(EP_2),(EP_3)$	$<$								
D	$\leq$	$\leq$	$=$	$\leq$	$\leq$	$\leq$	$<$		
(R_1)	$=$	$=$	$\leq$	$=$	$<$				
(R_2)	$=$	$=$	$\leq$	$=$	$=$	$<$			
(MF_2)	$=$	$=$	$=$	$=$	$=$	$=$	$<$		
(MF_3)	$=$	$=$	$=$	$=$	$<$				
$(MF_1),(EP_1)$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$=$	$<$
$\mathbf{T},\mathbf{I},\mathbf{O},(MF_4)$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$\leq$	$<$		

Figure 4.7: Variations des fonctions d'interprétation dans l'élimination des diséquations

- Dans les autres cas, $T'(s = t) = T'(s \neq t) = 2 * \max(\text{taille}(s), \text{taille}(t))$

$\phi_{3,8}(d)$ est le nombre de membres d'équations et de diséquations de d qui sont des variables

$\phi_{3,9}(d)$ est le nombre d'inconnues auxiliaires et de paramètres de $\mathcal{P}$.

Le tableau de la figure 4.7 résume les variations de ces fonctions par application des règles. Un signe $\leq$ signifie que, dans certaines situations, la fonction est strictement décroissante et dans d'autres elle est constante. Un signe $<$ (resp. $=$) signifie que, si les fonctions d'interprétation des colonnes précédentes n'ont pas strictement décrû, alors celle de la colonne où apparait $<$ (resp. $=$) décroit strictement (resp. reste constante).

Nous ne justifions pas les résultats de cette figure. Une vérification analogue à celle des preuves des théorèmes de terminaison du chapitre 3 peut être effectuée sans trop de difficulté.

Comme dans le chapitre 3, la règle (Ex_1) ne figure pas car elle fait croitre temporairement Φ_3. Mais, comme précédemment, on peut extraire de toute chaine infinie de transformation une sous-suite strictement décroissante par Φ_3. Ce qui permet de prouver la terminaison. □

Le théorème qui suit établit la complétude de $\mathcal{R}_3$ vis-à-vis des problème d'unification, si *l'on restreint l'ensemble des formes initiales aux preoblèmes SED.* Cette restriction sera discutée dans le paragraphe suivant. Montrons tout d'abord que les propriétés EU et SED sont conservées par transformation:

Lemme 4.11 *Si $\mathcal{P}$ est SED et $\mathcal{P} \mapsto_{\mathcal{R}_3} \mathcal{P}'$, alors $\mathcal{P}'$ est SED.*

Preuve

Il suffit de vérifier qu'aucune règle ne permet d'introduire d'équation dans les disjonctions. □

Lemme 4.12 *Si $\mathcal{P}$ a la propriété EU et que $\mathcal{P} \mapsto_{\mathcal{R}_3} \mathcal{P}'$, alors $\mathcal{P}'$ a la propriété EU.*

Preuve
Le résultat est trivial lorsque $\mathcal{P} \mapsto_R \mathcal{P}'$ par une règle R qui est fortement adéquate, puisqu'alors $\mathcal{P} \approx \mathcal{P}'$. Il suffit donc de remarquer que la propriété EU est conservée par application de (Ex_1) puisque cette règle consiste seulement à ajouter une équation au problème. □

Théoreme 4.13 *Si $\mathcal{P}$ est SED, alors $\mathcal{P}$ a la propriété EU ssi ses formes irréductibles pour $\mathcal{R}_3$ sont des problèmes d'unification.*

Preuve
Nous allons montrer que les formes irréductibles d'un problème SED satisfont les hypothèses du théorème 4.9. les lemmes 4.11 et 4.12 permettent alors d'obtenir la conclusion du théorème.

Soit donc $\mathcal{P}$ un problème SED qui ne satisfait pas les hypothèses du théorème 4.9. Par hypothèse, $\mathcal{P} \equiv \exists \vec{w}, \forall \vec{y} : t_1 = u_1 \wedge \ldots \wedge t_n = u_n \wedge d_1 \wedge \ldots \wedge d_m$ où les d_i sont de disjonctions de diséquations. Un certain nombre de cas se présentent alors:

1. **$\mathcal{P}$ contient des occurrences de paramètres**
   $\mathcal{P}$ est alors réductible par un raisonnement analogue à celui de la section 3.4.

2. **Il existe un indice i tel que ni u_i ni t_i n'appartient à $\mathcal{I}$**
   On peut alors appliquer (D_1) ou (I_1) si ni t_i ni u_i n'est une variable. Supposons donc que $t_i \equiv z_i \in X$ (et que le problème ne contient pas de paramètre, sans quoi nous tombons dans le cas précédent). z_i est donc une variable auxiliaire. Si $z_i \in Var(u_i)$, alors une des règles $(T_1), (O_1)$ est applicable. Si maintenant $z_i \notin Var(u_i)$ et a une autre occurrence dans $\mathcal{P}$, on peut appliquer (R_1) (à moins que u_i ne soit elle même une variable résolue; nous écartons ce cas par raison de symétrie). Il ne reste plus alors à envisager que le cas $w_i = u_i$ où w_i est une inconnue auxiliaire résolue. Mais alors la règle (MF_2) est applicable.

Nous pouvons désormais supposer que:

$$\mathcal{P} \equiv \exists \vec{w} : x_1 = t_1 \wedge \ldots \wedge x_n = t_n \wedge d_1 \wedge \ldots \wedge d_m$$

où $x_1, \ldots, x_n \in \mathcal{I}$.

3. **Il existe un indice i tel que x_i a plus d'une occurrence dans $\mathcal{P}$**
   L'une des règles $(R_1), (O_1), (T_1)$ est applicable.

4. **$\mathcal{I} \neq \{x_1, \ldots, x_n\}$**
   Cela signifie qu'on a l'inclusion stricte $\{x_1, \ldots, x_n\} \subset \mathcal{I}$. Si $x \in \mathcal{I} - \{x_1, \ldots, x_n\}$, - ou bien $x \notin Var(\mathcal{P})$ - ou bien x n'est pas presque résolue dans $\mathcal{P}$ - ou bien il existe dans $\mathcal{P}$ une équation $x_1 = x$. Dans tous les cas, la règle (MF_3) est applicable.

5. $Var(t_1, \ldots, t_n) \cap \mathcal{I} \neq \emptyset$
Cela signifierait que l'un des x_i n'est pas résolu et nous retombons donc dans le cas 3.

Posons maintenant $d_i \equiv z_{1,i} \neq u_{1,i} \vee \ldots \vee z_{k,i} \neq u_{k,i}$.

6. Il existe des indices i, j tels que $z_{i,j}$ n'est pas une variable
Alors il serait possible d'appliquer (I_2) ou (D_2).

7. Il existe des indices i, j tels que $u_{i,j} \in T(F)$
Il est alors possible d'appliquer (Ex_1).

8. Il existe des indices i, j tels que $z_{i,j}$ a plus d'une occurrence dans d_j
A cause du point 7, on peut supposer que $z_{i,j}$ n'a pas d'occurrence dans une diséquation $z_{i,j} \neq u$ où $u \in T(F)$. Par conséquent, si l'on ne peut appliquer (R_2) c'est que -ou bien $z_{i,j} \in Var(u_{i,j})$ et (T_2) ou (O_2) est applicable -ou bien il existe une variable presque résolue (soit z) ayant une occurrence dans d_j. Mais, à cause du point 3, une telle variable ne peut être qu'auxiliaire et, vu la forme générale de $\mathcal{P}$, elle ne pourrait apparaitre que dans une équation $x = z$ où $x \in \mathcal{I}$. Mais dans ce cas, z n'est pas presque résolue pour la nouvelle définition de variable presque résolue que nous avons donnée.

9. $Var(d_1, \ldots, d_n) \not\subseteq Var(t_1, \ldots, t_n)$
Si l'on suppose que nous ne nous trouvons dans aucun des cas précédents, le problème $\mathcal{P}$ étant SED, les 5 conditions d'application de la règle (MF_4) se trouvent remplies.

□

4.1.3 Commentaires

On peut se poser la question de la nécessité de l'hypothèse "$\mathcal{P}$ est SED" dans le théorème 4.13. Voici un premier contre-exemple, lorsqu'on omet cette hypothèse:

Exemple 4.1 $F = \{0 :\rightarrow \underline{s};\ f : \underline{s} \rightarrow \underline{s}\}$. Considérons alors:

$$\mathcal{P} \equiv \exists w, w', w'' :\ x = w \wedge x' = w' \wedge (w = 0 \vee w = f(w'') \vee w \neq w')$$

$\mathcal{P}$ est irréductible pour $\mathcal{R}_3$. Ce n'est pas un problème d'unification. Et pourtant, comme $\exists w, w'' : w = 0 \vee w = f(w'') \sim \top$, $\mathcal{P}$ est équivalent à un problème d'unification.

Cet exemple prouve que les règles que nous avons données sont insuffisantes pour assurer la complétude dans le cas général. On peut alors penser ajouter la règle:

$$(S)\quad \mathcal{P}[z = t[u_1] \vee \ldots \vee z = t[u_p]] \ \mapsto\ \exists w, \mathcal{P}[x = t[w]]$$

Si $\{u_1, \ldots, u_p\} = \{f(\vec{w}), f \in F\}$ où les variables de $\vec{w}$ sont des variables auxiliaires du problème.

Cette règle seraità peu de choses près la négation de la règle (Ex_1)[4]. Elle est correcte et adéquate lorsque $\mathcal{A} = T(F)$. L'exemple ci-dessous en illustre une (autre) utilisation:

Exemple 4.2 F est comme dans l'exemple précédent et

$$\mathcal{P} \equiv \exists w, w_1, w_2 : x_1 = w_1 \wedge x_2 = w_2 \wedge (w_1 = 0 \vee w_2 = 0 \vee w_1 = f(f(w)) \vee w_1 \neq f(w_2))$$

$$\begin{array}{ll} & \mathcal{P} \\ \mapsto_{R_2} & \exists w, w_1, w_2 : x_1 = w_1 \wedge x_2 = w_2 \wedge (f(w_2) = 0 \vee w_2 = 0 \vee w_1 = f(f(w)) \vee w_1 \neq f(w_2)) \\ \mapsto_{I} & \exists w, w_1, w_2 : x_1 = w_1 \wedge x_2 = w_2 \wedge (w_2 = 0 \vee w_1 = f(f(w)) \vee w_1 \neq f(w_2)) \\ \mapsto_{R_2} & \exists w, w_1, w_2 : x_1 = w_1 \wedge x_2 = w_2 \wedge (w_2 = 0 \vee f(w_2) = f(f(w)) \vee w_1 \neq f(w_2)) \\ \mapsto_{D_2} & \exists w, w_1, w_2 : x_1 = w_1 \wedge x_2 = w_2 \wedge (w_2 = 0 \vee w_2 = f(w) \vee w_1 \neq f(w_2)) \\ \mapsto_{S} & \exists w, w_1, w_2 : x_1 = w_1 \wedge x_2 = w_2 \end{array}$$

Conclusion
Il semble ainsi que la règle (S) permette d'éviter la condition SED tout en préservant la complétude. Nous n'avons pas pu trouver de contre-exemple. A l'inverse, nous n'avons pas pu prouver (jusqu'à présent) ce résultat de complétude. La difficulté essentielle réside dans la généralisation du théorème 4.9 aux problèmes qui, éventuellement, ne sont plus SED (mais sont irréductibles pour (S)). C'est pourquoi nous formulons la:

Conjecture 4.1 *Le système formé des règles du chapitre 3 et de la règle (S) permet d'éliminer la négation d'un problème équationnel (cahque fois que c'est possible).*

4.2 Formes résolues dans les arbres rationnels

L'algèbre $RT(F)$ des arbres rationnels construits sur la signature F a été introduite dans la section 2.4.3. En ce qui concerne la transformation des problèmes équationnels dans cette algèbre, notons que les règles **I**,**D**,**O**, $(EP_3), (EP_4), (MF_4), (Ex_2)$ n'ont pas été prouvées correctes dans $RT(F)$.

Ce que nous nous proposons donc de faire ici est de montrer tout d'abord que certaines de ces règles sont adéquates dans $RT(F)$ puis de montrer comment les autres peuvent être remplacées par des règles qui assurent la complétude.

Tout d'abord, **I**,**D**,$(EP_3), (EP_4), (Ex_2)$ sont, en fait, fortement adéquates dans $RT(F)$. La série des lemmes suivants a pour but d'amener à ce résultat.

Lemme 4.14 *Si $t \in RT(F)$ possède un sous arbre strict de même sorte que lui, alors t est infinitaire.*

Preuve
Supposons que $p \in Pos(t)$, $p \neq \epsilon$ et $sorte(t/p) = \underline{s} = sorte(t)$. Soit alors t_0 le terme obtenu en remplaçant tous les sous-arbres de t de profondeur $|p|$ par des variables (de sorte appropriée). Soit enfin σ une substitution fermée de domaine $Var(t_0)$. (Une telle

[4]Notre ensemble de règles serait alors "fermé par négation". On peut imaginer que cette propriété assurant une certaine homogénéité à l'ensemble des règles permette d'obtenir des résultats de complétude supplémentaires.

substitution existe puisque $T(F)$ est supposé contenir au moins un terme de chaque sorte).

On construit alors par récurrence la suite de termes fermés suivante: $t_{n+1} = (t[t_n]_p)\sigma$. Les termes t_i sont dans $T(F)$ et ont pour sorte $\underline{s}$. De plus, ils sont tous distincts puisque de profondeurs distinctes. Cela prouve que $\underline{s}$ est infinitaire.□

Lemme 4.15 *Si $t \in RT(F)$ est de sorte à support fini dans $RT(F)$, alors $t \in T(F)$.*

Preuve
Si $t \in RT(F)$ contient un sous arbre à la position p dont la sorte a un support infini (soit C) dans $RT(F)$, alors on obtient une infinité d'arbres distincts de sorte $sort(t)$ en remplaçant le sous-arbre de t de position p par un élément de C.

Par suite, si t est de sorte à support fini dans $RT(F)$, t ne contient pas de sous-arbre de sorte à support infini dans $RT(F)$. Si $i_1 i_2 \ldots i_n \in Pos(t)$, les sous-termes $t_1 \equiv t/i_1, \ldots, t_n \equiv t/i_1 \ldots i_n$ sont ainsi de sorte à support fini dans $RT(F)$. De plus, d'après le lemme 4.14, $t_1, \ldots, t_n$ sont de sortes distinctes. Si bien que $n \leq |S|$. Toute position de t étant de longueur bornée, on en déduit que $t \in T(F)$. □

Lemme 4.16 *Si $t \in RT(F) - T(F)$, alors t est infinitaire.*

Preuve
Soit $t \in RT(F) - T(F)$. Soit $i_m, m \geq 1$ une suite infinie d'entiers tels que $p_m = i_1 \cdot i_2 \cdot \ldots \cdot i_m$ soit une position de t pour tout m. Comme S est supposé fini, il existe deux indices m_1 et m_2 tels que $sorte(t/p_{m_1}) = sorte(t/p_{m_2})$. D'après le lemme 4.14, $\underline{s} = sorte(t/p_{m_1})$ est infinitaire. Soit alors t_0 le terme obtenu en remplaçant dans t tout sous-arbre de profondeur $1 + |p_{m_1}|$ par une variable de sorte appropriée. Soit enfin σ une substitution fermée de domaine $Var(t_0)$. On obtient alors une suite infinie de termes fermés distincts de sorte $sorte(t)$ en remplaçant dans $t_0\sigma$ le sous-terme de position p_{m_1} par un terme de sorte $\underline{s}$. Ce qui prouve que t est infinitaire. □

Lemme 4.17 *Si $t \in RT(F)$ est de sorte à support infini dans $RT(F)$ si et seulement si t est infinitaire.*

Preuve
L'un des sens de l'implication est une conséquence du lemme 4.15. Si maintenant $\underline{s} \in S$ a un support infini (soit C) dans $RT(F)$, alors -ou bien $C \subseteq T(F)$ et le résultat est trivial -ou bien C contient au moins un arbre rationnel et $\underline{s}$ est infinitaire d'après le lemme 4.16. □

Lemme 4.18 *Soit $\mathcal{P}$ une conjonction de diséquations non triviales dont les variables ont un support infini dans $RT(F)$. Alors $\mathcal{P}$ a au moins une solution dans $RT(F)$.*

Preuve
Cela résulte du lemme 3.7. En effet, si toute variable de $\mathcal{P}$ a un support infini dans $RT(F)$, d'après le lemme 4.17, elles sont infinitaires. Il en résulte que $\mathcal{P}$ a au moins une solution

dans $T(F)$ et donc dans $RT(F)$. □

Ce résultat se généralise au cas où ces diséquations ont pour membres des éléments de $RT(F,X)$. $RT(F,X)$ est simplement l'algèbre des arbres rationnels contruite sur la signature $F \cup X$, tous les symboles de X étant considérés comme des constantes. Nous parlerons le cas échéant de *RT-problème équationnel* lorsque les membres des équations et diséquations peuvent être des éléments de $RT(F,X)$ et non plus seulement des termes. Notre propos n'est pas d'étudier les RT-problèmes. Ces notions ne sont introduites ici que pour plus de commodité dans l'expression des preuves. Les définitions de solution, validation,... s'étendent de manière évidente. Notons aussi que le théorème 2.8 reste correct lorsque l'on remplace les termes t_i par des arbres rationnels de $RT(F,X)$. Enfin le lemme 4.18 se généralise comme annoncé ci-dessus:

Lemme 4.19 *Si $\mathcal{P}$ est un RT-problème constitué d'une conjonction de diséquations non triviales dont les variables ont un support infini dans $RT(F)$, alors $\mathcal{P}$ a au moins une solution dans $RT(F)$.*

Preuve

Il suffit en fait de reprendre la preuve du lemme 3.7. Celle-ci repose sur la seule propriété qu'une équation non triviale contenant une seule variable a au plus une solution et utilise ensuite le fait que les sortes sont infinitaires. Or cette propriété des équations à une variable reste vraie pour les RT-équations à une variable. On peut en effet prouver cette dernière propriété de la façon suivante: étant donné une équation $u = v$ où u et v sont distincts et $u, v \in RT(F, \{x\})$, simplifier (en utilisant incompatibilités et décompositions) cette équation tant que c'est possible. Cette transformation termine puisque les arbres u et v sont distincts: il y a une position à laquelle ils sont distincts. On obtient alors, s'il n'y a pas eu incompatibilité, une équation $x = t$ qui possède une unique solution d'après le théorème 2.8. □

Proposition 4.20 *Les règles* **I**, **D**, (EP_3), (EP_4) *et* (MF_4) *sont fortement adéquates dans $RT(F)$*

Preuve

En ce qui concerne **I** et **D**, ce résultat est trivial. En ce qui concerne la règle (EP_4), la seule difficulté est de voir que l'on obtient bien un problème équationnel après application de cette règle. En effet, à priori, $t_1, \ldots, t_n \in RT(F)$. Mais le lemme 4.16 nous assure que, si $\{t_1, \ldots, t_n\}$ est le support de $\underline{s}$ dans $RT(F)$, alors $t_1, \ldots, t_n \in T(F)$. Par conséquent les remplacements de la règle (EP_4) ont bien un sens.

D'autre part, les forte adéquations des règles (EP_3) et (MF_4) se déduisent l'une de l'autre comme il a été déjà remarqué dans le chapitre 3. Il reste donc seulement à prouver la forte adéquation de (EP_3).

Par le lemme 2.5 et en prenant la négation du problème, nous avons seulement à prouver que, si σ est une $RT(F)$-substitution quelconque sur les inconnues du problème, alors $z_1\sigma \neq u_1\sigma \wedge \ldots \wedge z_n\sigma \neq u_n\sigma$ a au moins une solution dans $RT(F)$ (les inconnues de ce problème étant les variables de $\vec{y}$). Mais aucune des diséquations du problème n'est

$$(RT_1) \quad \forall \vec{y} : P \wedge (y_1 \neq t_1 \vee \ldots \vee y_n \neq t_n \vee y_{n+1} = t_{n+1} \ldots \vee y_{n+m} = t_{n+m} \vee d) \quad \mapsto \vec{y} : P \wedge d$$

Si

1. d est une disjonction d'équations et de diséquations ne contenant pas de paramètre
2. $y_1, \ldots, y_n$ sont des paramètres distincts
3. $y_{n+1}, \ldots, y_{n+m}$ sont des paramètres
4. pour tout i, $n+1 \leq i \leq n+m$, y_i est infinitaire et $y_i \not\equiv t_i$
5. Les trois ensembles $\{y_1, \ldots, y_n\}$, $\{y_{n+1}, \ldots, y_{n+m}, t_{n+1}, \ldots, t_{n+m}\}$, $\{t_1, \ldots, t_m\}$ sont disjoints.

Figure 4.8: Elimination des paramètres dans $RT(F)$

triviale puisque, par le contrôle imposé à (EP_3), d'une part $z_i = u_i$ contient au moins une occurrence de paramètre, d'autre part $z_i \not\equiv u_i$. Enfin, les paramètres apparaissant dans ce problèmes sont infinitaires, de nouveau à cause du contrôle imposé. Le lemme 4.19 s'applique donc et nous fournit le résultat souhaité.□

Les tests d'occurrence étant incorrects lorsque $\mathcal{A} = RT(F)$, il nous faut donner d'autres règles permettant de les remplacer. Sans eux, il n'y a en effet plus complétude:

$$\forall \vec{y} : (y \neq f(y) \vee d)$$

est par exemple irréductible.

Une première étape permettant de réduire de tels problèmes consiste en l'introduction d'une nouvelle règle qui généralise (EP_3) en prenant en considération à la fois équations et diséquations contenant des paramètres. Cette règle est donnée dans la figure 4.8. La proposition suivante donne les résultats d'adéquation voulus:

Proposition 4.21 *La règle (RT_1) est fortement adéquate lorsque $\mathcal{A} = RT(F)$.*

Preuve

D'après le lemme 2.5, il suffit de prouver la forte adéquation de la règle

$$\forall \vec{y} : y_1 \neq t_1 \vee \ldots \vee y_n \neq t_n \vee y_{n+1} = t_{n+1} \vee \ldots \vee y_{n+m} = t_{n+m} \mapsto \bot$$

Autrement dit, il suffit de prouver que le problème en membre gauche ci-dessus n'a aucune solution.

On raisonne par l'absurde et on suppose que σ est une solution du membre gauche. Nous allons exhiber une substitution sur $\vec{y}$ qui valide

$$\mathcal{Q} \equiv y_1 = t_1\sigma \wedge \ldots \wedge y_n = t_n\sigma \wedge y_{n+1} \neq t_{n+1}\sigma \wedge \ldots \wedge y_{n+m} \neq t_{n+m}\sigma$$

ce qui amènera la contradiction souhaitée[5].

La partie équationnelle de $\mathcal{Q}$ a au moins une solution θ_0 dans $RT(F)$, si l'on considère $y_1, \ldots, y_n$ comme inconnues. (C'est une conséquence du théorème 2.8 et des conditions 2 et 5 d'application de la règle). Maintenant, si l'on applique θ_0 à la partie "diséquationnelle" de $\mathcal{Q}$, on obtient le problème

$$\mathcal{Q}_0 \equiv y_{n+1} \neq t_{n+1}\sigma\theta_0 \wedge \ldots \wedge y_{n+m} \neq t_{n+1}\sigma\theta_0$$

puisque $y_{n+i}\theta_0 \equiv y_{n+i}$ d'après la condition 5 imposée à l'application de la règle. Mais $\mathcal{Q}_0$ possède au moins une solution θ_1 dans $RT(F)$ d'après le lemme 4.19. Pour terminer, $\theta = \theta_0\theta_1$ valide $\mathcal{Q}$. □

Malheureusement le système obtenu est encore insuffisant pour obtenir la complétude. En effet, un problème comme

$$\mathcal{P} \equiv \forall y : y = f(y) \vee y \neq f(f(y))$$

reste irréductible puisque la règle (RT) ne peut être utilisée que lorsque l'ensemble des paramètres qui sont membre d'une équation est disjoint de l'ensemble des paramètres qui sont membre d'une diséquation. La règle (F_4) quant à elle ne peut pas non plus être utilisée puisqu'elle réclame l'inégalité taille-param$(t) \leq$ taille-param(u) qui n'est pas satisfaite ici. De plus, il n'est pas possible d'affaiblir l'une de ces deux conditions d'application. Par exemple, si l'on autorise la fusion (F_4) sans l'inégalité sur les tailles des positions des paramètres, le problème $\mathcal{P}$ ci-dessus est transformé en:

$$\forall y : f(f(y)) = f(y) \vee y \neq f(f(y))$$

qui est à nouveau transformé en $\mathcal{P}$ par décomposition.

Ainsi, si l'on ne considère que les règles déjà énoncées, l'ensemble de règles n'est complet que si nous nous restreignons aux problèmes SED.

Il est néanmoins possible de traiter le cas général en utilisant une méthode analogue à celle de A. Colmerauer [Col82]. Les règles de la figure 4.9 sont en effet des transcriptions dans notre formalisme de celles de A. Colmerauer. Elles présentent l'inconvénient d'utiliser des conditions d'application "sémantiques" au contraire de toutes les règles que nous avions données jusqu'ici. C'est pourquoi elles n'ont pas été énoncées plus tôt[6].

Proposition 4.22 *Les règles de la figure 4.9 sont fortement adéquates pour toute F-algèbre $\mathcal{A}$.*

[5] $\mathcal{Q}$ est ici un RT-problème

[6] Nous avons donné dans cette figure aussi bien les règles que leur négation, mais nous ne nous servirons que des transformations de diséquations. C'est pourquoi les conditions d'aplication ont été exprimées sous cette forme. Elles pourraient l'être sous la forme de systèmes d'équations comme dans [Col82].

$$(RT_2) \quad t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t = u \quad \mapsto t_1 \quad \neq u_1 \vee \ldots \vee t_n \neq u_n$$
$$(RT_2') \quad t_1 = u_1 \wedge \ldots \wedge t_n = u_n \wedge t \neq u \quad \mapsto \quad t_1 = u_1 \wedge \ldots \wedge t_n = u_n$$

Si $t_1 \neq u_1 \ldots \vee t_n \neq u_n \vee t \neq u \approx_{\mathcal{I},\mathcal{A}} \top$ où $\mathcal{I} = Var(t_1, \ldots, t_n, u_1, \ldots, u_n, t, u)$.

$$(RT_3) \quad t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t = u \quad \mapsto \quad \top$$
$$(RT_3') \quad t_1 = u_1 \wedge \ldots \wedge t_n = u_n \wedge t \neq u \quad \mapsto \quad \bot$$

Si $t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t \neq u \approx_{\mathcal{I},\mathcal{A}} t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n$ où $\mathcal{I} = Var(t_1, \ldots, t_n, u_1, \ldots, u_n, t, u)$.

$$(RT_4) \quad t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t = u \quad \mapsto \quad t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee \exists t_1' = u_1' \wedge \ldots \wedge t_m' = u_m')$$
$$(RT_4') \quad t_1 = u_1 \wedge \ldots \wedge t_n = u_n \wedge t \neq u \quad \mapsto \quad t_1 = u_1 \wedge \ldots \wedge t_n = u_n \wedge (t_1' \neq u_1' \vee \ldots \vee t_m' \neq u_m')$$

Si $t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t \neq u \approx_{\mathcal{I},\mathcal{A}} t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t_1' \neq u_1' \vee \ldots \vee t_m' \neq u_m'$ et $\mathcal{I} = Var(t_1, \ldots, t_n, u_1, \ldots, u_n, t, u)$

Figure 4.9: Règles de transformation utilisant des conditions "sémantiques"

Preuve

Il suffit bien sûr de prouver ce résultat pour les règles $(RT_2), (RT_3), (RT_4)$.

Forte adéquation de (RT_2)

Une des inclusions est immédiate. Il reste à montrer que toute $\mathcal{A}$-substitution σ qui valide le membre gauche valide aussi le membre droit. Mais une telle substitution σ, par hypothèse (contrôle) valide à la fois $t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t = u$ et $t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t \neq u$. Donc elle valide aussi leur conjonction

$$t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee (t = u \wedge t \neq u) \equiv t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n$$

Forte adéquation de (RT_3)

par hypothèse

$$t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t \neq u \approx_{\mathcal{I},\mathcal{A}} t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n$$

On en déduit, par propriété de monotonie,

$$t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t \neq u \vee t = u \approx_{\mathcal{I},\mathcal{A}} t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t = u$$

Ce qui donne le résultat voulu.

Correction de (RT_4)

Soit σ une $\mathcal{A}$-substitution validant $t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t = u$. Deux cas se présentent:

- ou bien il existe un indice i tel que $t_i\sigma \neq_{\mathcal{A}} u_i\sigma$ et σ valide trivialement le membre droit
- ou bien, pour tout i, $t_i\sigma =_{\mathcal{A}} u_i\sigma$. Alors $t\sigma =_{\mathcal{A}} u\sigma$. Donc σ ne valide pas $t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t \neq u$. Par suite (à cause des hypothèses d'application de la règle), σ ne valide pas non plus $t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t'_1 \neq u'_1 \vee \ldots \vee t'_m \neq u'_m$. Ce qui prouve en particulier que, pour tout j, $t'_j\sigma =_{\mathcal{A}} u'_j\sigma$. σ valide donc le membre droit.

Preservation de (RT_4)

Un raisonnement analogue au précédent conduit au résultat.

□

Ces règles posent un problème : celui de savoir si elles sont applicables ou non. Les conditions données font en effet intervenir l'équivalence des problèmes, or nous n'avons pas encore de moyen d'en décider. Nous allons donc renforcer ces conditions de façon à ce qu'elles puissent être facilement vérifiées. Il nous faut auparavant énoncer quelques résultats permettant justement d'établir des cas d'équivalence de problèmes. Ces résultats ne sont pas nouveaux: ils ne sont que la transcription dans notre formalisme de résultats donnés par A. Colmerauer dans [Col82].

Lemme 4.23 [Col82]
Soient $e \equiv z_1 = u_1 \wedge \ldots \wedge z_n = u_n$ et $e' \equiv z_1 = v_1 \wedge \ldots \wedge z_n = u_n$ deux problèmes dans lesquels $z_1, \ldots, z_n$ sont des variables distinctes. Soit $\mathcal{A} = T(F)$ ou $RT(F)$. Si $\mathcal{S}(\mathcal{A}, e, \mathcal{I})$ est non vide et contenu dans $\mathcal{S}(\mathcal{A}, e', \mathcal{I})$, alors ces deux ensembles de solutions sont égaux.

Preuve

Notons $V = Var(u_1, \ldots, u_n) - \{z_1, \ldots, z_n\}$ et $V' = Var(v_1, \ldots, v_n) - \{z_1, \ldots, z_n\}$. Alors $V' \subseteq V$. En effet, supposons que ce n'est pas le cas. Soient alors $z \equiv v_i/p$ et $z \notin V$. Soit σ une $\mathcal{A}$-solution de e. Soit enfin $t \in \mathcal{A}$ tel que $t \neq_{\mathcal{A}} z_i\sigma/p$. Alors $v_i\{z \rightarrow t\}\sigma \neq_{\mathcal{A}} z_i\sigma$, ce qui prouve que $\{z \rightarrow t\}\sigma$ n'est pas une solution de e'. C'est absurde car cette substitution est solution de e.

Soit maintenant σ une solution quelconque de e'. Nous allons montrer que c'est une solution de e. Soit σ' la restriction de σ à V'. $e\sigma'$ admet au moins une solution. En effet, dans le cas où $\mathcal{A} = RT(F)$, c'est une conséquence du théorème 2.8 et dans le cas où $\mathcal{A} = T(F)$, soit $z_1 = u'_1 \wedge \ldots \wedge z_n = u'_n$ la forme irréductible par remplacements de e. La substitution $\{z_1 \rightarrow u'_1; \ldots; z_n \rightarrow u'_n\}\sigma'\theta$ est solution de e, si θ est une substitution quelconque de domaine $V - V'$. Donc $\{z_1 \rightarrow u'_1\sigma'\theta; \ldots; z_n \rightarrow \sigma'\theta\}\theta$ est solution de $e\sigma'$.

Mais, d'après le théorème 2.8, $e'\sigma'$ a une unique solution dans $RT(F)$ (et donc au plus une dans $T(F)$). Comme $\mathcal{S}(\mathcal{A}, e\sigma', \mathcal{I}) \subseteq \mathcal{S}(\mathcal{A}, e'\sigma', \mathcal{I})$, que l'ensemble de gauche a au moins un élément et celui de droite au plus un, il sont nécessairement égaux. σ est donc solution de e. □

Corollaire 4.24 *Soient $d_1 \equiv z_1 \neq u_1 \vee \ldots \vee z_n \neq u_n \vee z \neq u$ et $d_2 \equiv z_1 \neq v_1 \vee \ldots \vee z_n \neq v_n \vee z'_1 \neq u'_1 \ldots \vee z'_m \neq u'_m$ deux problèmes équivalents (par rapport à $RT(F)$ et $\mathcal{I}$) tels*

que $z_1, \ldots, z_n, z'_1, \ldots, z'_m$ sont des variables distinctes, z est une variable et pour tout i, $z_i \not\equiv v_i$ et $z'_i \not\equiv u'_i$. Alors

$$d_1 \approx_{RT(F),\mathcal{I}} z_1 \neq u_1 \vee \ldots \vee z_n \neq u_n \vee z'_1 \neq u'_1 \vee \ldots \vee z'_m \neq u'_m$$

Preuve

Notons $e_1 \equiv z_1 = u_1 \wedge \ldots \wedge z_n = u_n$, $e_2 \equiv z_1 = v_1 \wedge \ldots \wedge z_n = v_n$ et $e_3 \equiv z'_1 = u'_1 \wedge \ldots \wedge z'_m = u'_m$. Permettons nous d'omettre aussi $RT(F)$ et $\mathcal{I}$ dans les ensembles de solutions. L'hypothèse est ainsi que $\mathcal{S}(e_1 \wedge z = u) = \mathcal{S}(e_2 \wedge e_3)$. Comme $\mathcal{S}(e_2 \wedge e_3)$ est un sous-ensemble d'à la fois $\mathcal{S}(e_1)$ et $\mathcal{S}(e_3)$, c'est un sous-ensemble de $\mathcal{S}(e_1 \wedge e_3)$. D'autre part, $\mathcal{S}(e_2 \wedge e_3)$ est non vide d'après le théorème 2.8. Par conséquent, d'après le lemme 4.23, $\mathcal{S}(e_2 \wedge e_3) = \mathcal{S}(e_1 \wedge e_3)$, ce qui est le résultat souhaité. □

Le système de règles $\mathcal{R}_4$ permettant d'aboutir à un résultat analogue à celui du chapitre 3 dans le cas où $\mathcal{A} = RT(F)$ est alors simplement constitué des règles du système $\mathcal{R}_2$ de la section 3.6 auquel on a retiré les tests d'occurrence et auquel on a ajouté les règles des figures 4.8 et 4.10[7].

On aura alors un résultat de terminaison et de complétude analogue à celui du chapitre 3. Donnons d'abord la définition d'une "définition contrainte" dans le cas où $\mathcal{A} = RT(F)$.

Définition 4.25 *Un* cycle de variable *est un système*

$$z_1 = z_2 \wedge \ldots \wedge z_{n-1} = z_n \wedge z_n = z_1$$

où $n \geq 1$ et $z_1, \ldots, z_n$ sont des variables.

Définition 4.26 *Lorsque $\mathcal{A} = RT(F)$, un problème équationnel est une* définition contrainte *s'il est égal à $\top$ ou $\bot$ ou bien de la forme*

$$\exists \vec{w} : z_1 = t_1 \wedge \ldots \wedge x_m = t_m \wedge z'_1 \neq t'_1 \wedge \ldots \wedge z'_n \neq t'_n$$

avec:

1. *$z_1, \ldots, z_m$ sont des inconnues distinctes*
2. *il n'y a pas de cycle de variables*
3. *pour tout indice i, x'_i est in finitaire et est distinct de t'_i*
4. *$\{z_1, \ldots, z_m\} \cap \{z'_1, \ldots, z'_n, t'_1, \ldots, t'_n\} = \emptyset$*

Théoreme 4.27 *$\mathcal{R}_4$ est à terminaison finie et les formes irréductibles pour $\mathcal{R}_4$ sont des définitions contraintes.*

Preuve

Nous n'allons pas refaire ici la preuve du théorème 3.20. Concernant la terminaison il suffit d'ailleurs de remarquer que les mêmes fonctions d'interprétation que celles du théorème

[7] La règle (RT_4) de la figure 4.10 est bien un cas particulier de la règle (RT_4) de la figure 4.9 d'après le corollaire 4.24.

$$
\begin{array}{lll}
(RT_2) & t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t = u & \mapsto t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \\
(RT_2') & t_1 = u_1 \wedge \ldots \wedge t_n = u_n \wedge t \neq u & \mapsto t_1 = u_1 \wedge \ldots \wedge t_n = u_n
\end{array}
$$

Si $t_1 \neq u_1 \ldots \vee t_n \neq u_n \vee t \neq u \mapsto^*_{\mathbf{D},\mathbf{F},\mathbf{I}} \top$.

$$
\begin{array}{lll}
(RT_3) & t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t = u & \mapsto \top \\
(RT_3') & t_1 = u_1 \wedge \ldots \wedge t_n = u_n \wedge t \neq u & \mapsto \bot
\end{array}
$$

Si $t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee t \neq u \mapsto_{\mathbf{D},\mathbf{F},\mathbf{I}} t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n$.

$$
\begin{array}{lll}
(RT_4) & z_1 \neq u_1 \vee \ldots \vee z_n \neq u_n \vee z = u & \mapsto z_1 \neq u_1 \vee \ldots \vee z_n \neq u_n \vee (z_1' = u_1' \wedge \ldots \wedge z_m' = u_m') \\
(RT_4') & z_1 = u_1 \wedge \ldots \wedge z_n = u_n \wedge z \neq u & \mapsto z_1 = u_1 \wedge \ldots \wedge z_n = u_n \wedge (z_1' \neq u_1' \vee \ldots \vee z_m' \neq u_m')
\end{array}
$$

Si

- $z_1 \neq u_1 \vee \ldots \vee z_n \neq u_n \vee z \neq u \mapsto_{\mathbf{D},\mathbf{F},\mathbf{I}} z_1 \neq v_1 \vee \ldots \vee z_n \neq v_n \vee z_1' \neq u_1' \vee \ldots \vee z_m' \neq u_m'$.
- $z_1, \ldots, z_n, z_1', \ldots, z_n'$ sont des variables distinctes
- pour tous i, j, $z_i \not\equiv v_i$ et $z_i' \not\equiv u_i'$

Figure 4.10: Règles de transformation dans $RT(F)$: contrôle

3.20 sont décroissantes par application des règles de $\mathcal{R}_4$. D'autre part, les nouvelles règles permettent effectivement d'asurer l'élimination des paramètres. La condition 4 dans la définition 4.26 est par ailleurs assurée par les règles $(RT'_2), (RT'_3), (RT'_4)$ tandis que la condition 2 est assurée par la règle de remplacement (R_1). □

Ce résultat entraine la complétude vis-à-vis des définitions contraintes:

Corollaire 4.28 *$\mathcal{R}_4$ est complet vis-à-vis de $\mathcal{F}_I$ ensemble de tous les problèmes équationnels, $\mathcal{F}_R$ ensemble des définitions contraintes et $\mathcal{A} = RT(F)$.*

Ces formes résolues, comme dans le cas de $T(F)$, possèdent la bonne propriété d'avoir au moins une solution dans $RT(F)$:

Proposition 4.29 *Les définitions contraintes (dans $RT(F)$) possèdent au moins une solution dans $RT(F)$.*

Preuve

C'est une conséquence du lemme 4.19. Le système de diséquations possède en effet une $RT(F)$-solution σ. Alors $z_1 = t_1\sigma \wedge \ldots \wedge z_m = t_m\sigma$ eest un RT-système qui a au moins une solution d'après le théorème 2.8. □

Enfin, de même que dans le cas de $T(F)$, on peut en déduire un résultat de décidabilité en calcul du premier ordre:

Corollaire 4.30 *La validité dans $RT(F)$ d'une formule du premier ordre dont le seul symbole de prédicat est $=$ est décidable.*

Autrement dit, on obtient un résultat analogue à celui de MJ. Maher [Mah88a]: les règles de décomposition et incompatibilités constituent une axiomatisation complète des arbres rationnels sur un alphabet fini.

Terminons cette section par un exemple de transformation de problème:

Exemple 4.3
On reprend ici l'exemple 2.4.3. $S = \{\underline{s}\}$, $F = \{0 \to \underline{s};\ g : \underline{s} \times \underline{s} \to \underline{s}\}$.

$$\mathcal{P} \equiv \forall y_1, y_2 : y_1 \neq g(y_1, x) \vee y_1 \neq g(y_1, y_2) \vee y_2 \neq 0 \vee y_1 = 0$$

On obtient alors (par exemple) la suite de transformations de la figure 4.11.

Cette transformation montre que $\mathcal{P} \approx_{RT(F),\{x\}} x \neq 0$.

4.3 Résolution dans $T(F, X)$

Nous nous interessons ici à nouveau à une autre algèbre: $T(F, X)$. Ou plutôt, nous nous intéressons à la résolution des problèmes équationnels dans $T(F)$, lorsque F est infini. (Ce

$$\begin{array}{lll}\mathcal{P} & \mapsto_{EP_2,EP_1} & \forall y_1 : y_1 \neq g(y_1,x) \vee y_1 \neq g(y_1,0) \vee y_1 = 0 \\ & \mapsto_{F_2} & \forall y_1 : y_1 \neq g(y_1,x) \vee g(y_1,x) \neq g(y_1,0) \vee y_1 = 0 \\ & \mapsto_{D_2} & \forall y_1 : y_1 \neq g(y_1,x) \vee y_1 \neq y_1 \vee x \neq 0 \vee y_1 = 0 \\ & \mapsto_{T_2} & \forall y_1 : y_1 \neq g(y_1,x) \vee x \neq 0 \vee y_1 = 0\end{array}$$

Mais, comme

$$y_1 \neq g(y_1,x) \vee y_1 \neq 0 \mapsto^*_{\mathbf{F,I,D}} \top$$

on peut appliquer RT_2:

$$\begin{array}{lll}\forall y_1 : y_1 \neq g(y_1,x) \vee x \neq 0 \vee y_1 = 0 & \mapsto_{RT_2} & \forall y_1 : y_1 \neq g(y_1,x) \vee x \neq 0 \\ & \mapsto_{RT_1} & \forall y_1 : x \neq 0 \\ & \mapsto_{EP_1} & x \neq 0\end{array}$$

Figure 4.11: Exemple de transformation pour $\mathcal{A} = RT(F)$

cas contient celui de $T(F,X)$). Heureusement, ce cas est plus simple que les autres.

La seule règle de transformation qui n'est plus adéquate lorsque F est infini est la règle d'explosion.

Si l'on souhaite seulement assurer la solubilité (existence d'une solution) comme dans le chapitre 3, il est inutile d'éliminer complètement les paramètres lorsque F est infini:

Lemme 4.31 *Supposons que F contient, pour chaque sorte $\underline{s} \in S$ une infinité de symboles dont le codomaine est $\underline{s}$. Alors les problèmes de la forme $\forall \vec{y} : z_1 \neq u_1 \wedge \ldots \wedge z_n \neq u_n$ où $z_i \notin \vec{y}$, $u_i \notin \vec{y}$ et, pour tout i, $z_i \not\equiv u_i$ ont toujours une solution dans $T(F)$.*

Preuve

Nous nous ramenons tout d'abord au cas où toute variable est dans $\{z_1,\ldots,z_m\} \cup \vec{y}$ en appliquant une substitution fermée θ quelconque de domaine $Var(u_1,\ldots,u_m) - (\{z_1,\ldots,z_n\} \cup \vec{y})$. On raisonne ensuite par récurrence sur $m = |\{z_1,\ldots,z_n\}|$:

- Si $m = 1$, le problème s'écrit $\forall \vec{y} : z_1 \neq u_1 \wedge \ldots \wedge z_1 \neq u_n$. On choisit alors t de sorte que $t(\epsilon)$ soit un symbole distinct de $u_1(\epsilon),\ldots,u_n(\epsilon)$ (ces termes ne peuvent être des variables). $\{z_1 \rightarrow t\}$ est une solution du problème.
- Supposons la propriété vraie pour $m-1$. On choisit t tel que $t(\epsilon)$ est un symbole distinct de $u_i(\epsilon)$ pour tous les u_i non variable. Appliquons ensuite $\{z_n \rightarrow t\}$ au problème. Celui-ci se simplifie et l'on peut appliquer l'hypothèse de récurrence. □

Les formes résolues dans le cas où F est infini peuvent ainsi être "plus grossières" que dans le cas où F est fini, tout en assurant la solubilité. Des constatations analogues

servent d'ailleurs de base aux résultats de Kunen [Kun87] et de Bürckert [Bur88].

Utilisons alors les règles de la section 3.6, à l'exception des règles **E**. Bien sûr, le système obtenu (Noté $\mathcal{R}_5$) est à terminaison finie puisque ce n'est qu'une "spécialisation" du système $\mathcal{R}_2$ qui est lui-même à terminaison finie (cf théorème 3.20). Montrons aussi qu'il est complet:

Théoreme 4.32 *Si F contient pour chaque sorte $\underline{s} \in S$ une infinité de symboles de codomaine $\underline{s}$, alors les formes irréductibles pour $\mathcal{R}_5$ ont au moins une solution dans $T(F)$.*

Preuve
Il suffit de remarquer que les formes irréductibles pour $\mathcal{R}_5$ sont de la forme:

$$\exists \vec{w}, \forall \vec{y} : z_1 = t_1 \wedge \ldots \wedge z_n = t_n \wedge d_1 \wedge \ldots \wedge d_n$$

où

- $\vec{y} \cap Var(z_1, t_1, \ldots, z_n, t_n) = \emptyset$
- $z_1, \ldots, z_n$ sont des variables qui n'ont qu'une occurrence dans le problème
- $d_i \equiv z_{1,i} \neq u_{1,i} \vee \ldots \vee z_{m_i,i} \neq u_{m_i,i} \vee e$ avec:
  - $z_{1,i}, \ldots, z_{m_i,i}$ sont des inconnues
  - $u_{1,i}, \ldots, u_{m_i,i}$ ne sont pas des paramètres
  - e est une disjonction d'équations
  - $m_i \geq 1$

On applique alors le lemme 4.31 en ne considérant qu'une diséquation par disjonction; la conjonction de diséquation ainsi obtenue ayant pour solution σ, on applique σ à la conjonction d'équations qui fournit à son tour une solution. □

Comme conséquence, nous avons, à nouveau, la décidabilité de la validité dans $T(F)$ d'une formule du premier ordre dont le seul symbole de prédicat est $=$, dans le cas où l'alphabet est infini. Terminons apr un exemple de transformation:

Exemple 4.4 Nous nous intéressons à la "résolution" dans $T(F, X)$ du problème

$$\mathcal{P} \equiv \forall y_1, y_2 : (g(y_1, y_2) \neq g(f(y_1), x_2) \vee x_1 = f(y_1)) \wedge (g(y_2, y_2) \neq g(0, x_2)) \vee x_1 = f(y_1))$$

On obtient successivement:

$$\begin{array}{ll} \mathcal{P} & \mapsto_{D_2, EP_2} \quad \forall y_1, y_2 : (x_2 \neq f(y_1) \vee x_1 = f(y_1)) \wedge (x_2 \neq 0 \vee x_1 = f(y_1)) \\ & \mapsto_{EP_1, EP_3} \quad \forall y_1 : x_2 \neq 0 \wedge (x_2 \neq f(y_1) \vee x_1 = f(y_1)) \end{array}$$

Ce dernier problème (bien que contenant des paramètres) est irréductible. On peut en effet exhiber des solution, par exemple en ne considérant pas l'équation $x_1 = f(y_1)$ et en choisissant pour x_2 un symbole de tête distinct de 0 et de g: pour tout $z \in X$ tel que $sort(z) = sort(x_2)$ et tout terme t, $\{x_1 \rightarrow t; x_2 \rightarrow z\}$ est une solution du problème.

4.4 Problèmes équationnels dans les OSA

Jusqu'à présent, nous nous placions dans le cadre des algèbres "multi-sortes" comme défini dans le chapitre 2. Nous nous intéressons ici à des algèbres dans lesquelles on autorise certaines relations entre les sortes (par exemple $nat < int$ signifiera que les entiers naturels sont aussi des entiers relatifs). Comme l'on montré Goguen et Meseguer [GM87b], l'utilisation de telles relations entre les sortes permet d'exprimer (simplement) des définitions qui ne peuvent l'être dans le cadre de la logique multi-sorte[8][Com88a]. Ces algèbres permettent d'exprimer les cas d'erreur [FGJM85,GM87b] et servent de base à la sémantique du langage OBJ [FGJM85,KKM88].

L'unification et les preuves dans les algèbres avec sortes ordonnées ont été largement étudiées. Citons [Wal85,Sch86,GM87a,Kir88,SNGM87,GKK88] entre autres. Nous nous intéressons ici à la disunification dans ces algèbres.

Commençons par rappeler les définitions fondamentales.

4.4.1 Definitions

Comme dans le cas multi-sorte, S est un ensemble de symboles de sortes et F un ensemble de symboles fonctionnels. S est de plus muni d'une relation d'ordre $\geq$ et F est muni d'une fonction τ qui associe à chaque symbole de F un sous-ensemble fini de S^+, ensemble des mots finis de longueur au moins 1 construits sur le vocabulaire S. On suppose que tous les éléments de $\tau(f)$ (appelés *profils de f*) ont la même longueur notée $|f|$. Ainsi, à la différence du cas multi-sorte, un même symbole fonctionnel peut avoir plusieurs profils. Lorsque $\underline{s}_1\underline{s}_2\ldots\underline{s}_n\underline{s} \in \tau(f)$, on note aussi $f : \underline{s}_1 \times \ldots \times \underline{s}_n \rightarrow \underline{s}$.

La relation d'ordre sur S est donnée par des *déclarations de sorte* de la forme $\underline{s}_1 < \underline{s}_2$. $\geq$ est alors défini comme la plus petite relation d'ordre satisfaisant ces déclarations.

Une *signature avec sortes ordonnées* est un quadruplet (S, F, SD, FD) formé d'un ensemble de sortes S, d'un ensemble de symboles fonctionnels F, d'un ensemble de déclarations de sorte SD et d'un ensemble de déclarations de profils FD. On écrira souvent $(S, F, \geq, FD)$ pour une signature avec sortes ordonnées, remplaçant SD par la relation d'ordre engendrée. La figure 4.12 donne un exemple de signature avec sortes ordonnées. Cet exemple correspond simplement à une définition des entiers relatifs ($\underline{int}$) possédant comme sous-sorte zéro ($\underline{zero}$), les entiers positifs ($\underline{pos}$), négatifs ($\underline{neg}$), strictement positifs ($\underline{spos}$) et strictement négatifs ($\underline{sneg}$).

Lorsque deux sortes $\underline{s}_1$ et $\underline{s}_2$ sont incomparables pour $\geq$ on note $\underline{s}_1 \bowtie \underline{s}_2$ (notation de C. Kirchner [Kir88]).

X est, comme dans le cas multi-sorte, un ensemble de variables, chacune étant munie d'une sorte. (On note $z : \underline{s}$ ou $sort(z) = \underline{s}$ pour "la variable z est de sorte $\underline{s}$). Un *terme* t de sorte $\underline{s}$ est toujours un arbre fini étiqueté par les symboles de F. Il doit vérifier les conditions de bonne formation suivante:

- si $t(p) = f$ et $|f| = n$, alors $p \cdot 1, \ldots, p \cdot n$ sont des positions de t (et ce sont les seules qui sont des suffixes de p et qui ont pour longueur $|p| + 1$) et il existe un profil $f : \underline{s}_1 \times \ldots \times \underline{s}_n \rightarrow \underline{s}'$ de f tel que t/p est une terme de sorte $\underline{s}'$ et, pour tout i, $t/p \cdot i$ est un terme de sorte $\underline{s}_i$.

[8] A moins d'utiliser des opérateurs cachés cf chapitre 6.

$S = \{\underline{zero}, \underline{pos}, \underline{neg}, \underline{spos}, \underline{sneg}, \underline{int}\}$

Déclarations de sorte :

$$\underline{zero} < \underline{pos} < \underline{int}$$
$$\underline{zero} < \underline{neg} < \underline{int}$$
$$\underline{spos} < \underline{pos} \text{ et } \underline{sneg} < \underline{neg}$$

$F = \{0, succ, pred, +\}$

Déclarations de profil :

$$\begin{array}{rrcl}
0: & & \rightarrow & \underline{zero} \\
succ: & \underline{pos} & \rightarrow & \underline{spos} \\
 & \underline{int} & \rightarrow & \underline{int} \\
pred: & \underline{neg} & \rightarrow & \underline{sneg} \\
 & \underline{int} & \rightarrow & \underline{int} \\
+: & \underline{pos} \times \underline{spos} & \rightarrow & \underline{spos} \\
 & \underline{spos} \times \underline{pos} & \rightarrow & \underline{spos} \\
 & \underline{sneg} \times \underline{neg} & \rightarrow & \underline{sneg} \\
 & \underline{neg} \times \underline{sneg} & \rightarrow & \underline{sneg} \\
 & \underline{int} \times \underline{int} & \rightarrow & \underline{int}
\end{array}$$

Figure 4.12: Exemple de signature avec sortes ordonnées

- si $t(p) \in X$, alors t/p est un terme de sorte $sort(t(p))$.

Lorsque Σ est une signature avec sortes ordonnées, $T(\Sigma, X)$ désigne, comme dans le cas multi-sorte, l'algèbre des termes construits sur la signature Σ et l'ensemble de variables X. $T(\Sigma)$ est une abréviation de $T(\Sigma, \emptyset)$. Comme précedemment, nous supposerons qu'il existe dans $T(\Sigma)$ au moins un terme de chaque sorte

Précisons maintenant quels sont les "modèles" d'une signature avec sortes ordonnées.

Définition 4.33 [SNGM87,Kir88] *Soit Σ une signature avec sortes ordonnées. Une Σ-algèbre $\mathcal{A}$ consiste en:*

- *Un ensemble $C_{\mathcal{A}}$ appelé* support *de $\mathcal{A}$*
- *Pour chaque $\underline{s} \in S$, un sous-ensemble $\underline{s}_{\mathcal{A}}$ de $C_{\mathcal{A}}$*
- *Pour chaque $f \in F$, une application $f_{\mathcal{A}}$ de $D_{f,\mathcal{A}}$ dans $C_{\mathcal{A}}$, où $D_{f,\mathcal{A}}$ est un sous-ensemble de $C_{\mathcal{A}}^{|f|}$ appelé* domaine *de f.*

tels que

- $\bigcup_{\underline{s} \in S} \underline{s}_{\mathcal{A}} = C_{\mathcal{A}}$
- *si $\underline{s} < \underline{s}'$ est une déclaration de sorte de Σ, alors $\underline{s}_{\mathcal{A}} \subseteq \underline{s}'_{\mathcal{A}}$*
- *si $f : \underline{s}_1 \times \ldots \times \underline{s}_n \rightarrow \underline{s}$ est une déclaration de profil et que, pour tout i, $a_i \in \underline{s}_{i,\mathcal{A}}$, alors $(a_1, \ldots, a_n) \in D_{f,\mathcal{A}}$ et $f_{\mathcal{A}}(a_1, \ldots, a_n) \in \underline{s}_{\mathcal{A}}$*

Une algèbre avec sortes ordonnées sera notée en abrégé OSA (pour "Order Sorted Algebra").

Cette définition de G. Smolka [SNGM87] a été reprise par C. Kirchner [Kir88]. Elle est différente de celle de Goguen et Meseguer [GM87a] car f n'est pas interprété comme *plusieurs* fonctions (une par profil) coincidant sur l'intersection des domaines, mais comme une seule fonction partielle. Cette définition nous semble plus naturelle et plus simple (il n'est plus nécessaire de donner des conditions de coincidence). Elle présente cependant l'inconvénient de ne pas étendre le cas multi-sorte (cf [GM87a]). De toutes façons, ce choix n'est pas crucial pour la suite car les deux définitions coincident dans le cas dans lequel nous nous placerons.

Bien sûr, $T(\Sigma, X)$ est une Σ-algèbre. Pour donner un autre exemple, avec la signature de la figure 4.12,

Exemple 4.5 Une Σ-algèbre est donnée par l'interprétation: $\mathcal{A} = Q$, ensemble des nombres rationnels, $\underline{zero}_Q = \{0\}$, $\underline{pos}_Q$ ensemble des rationels positifs ou nuls, $\underline{spos}$, ensemble des rationnels strictement positifs (même chose pour les négatifs), $\underline{int}_Q = Q$ et où *succ* et *pred* sont interprétés comme la fonction qui associe à x $2x + 1$ et $+$ est interprété comme l'addition.

Il nous faudra parfois nous restreindre à des signatures *régulières*. Cette hypothèse est asez naturelle et est effectuée par la plupart des auteurs [GM87a,SNGM87,Kir88,GKK88].

Définition 4.34 [GM87a] *La signature* Σ *est* régulière *ssi tout terme* $t \in T(F, X)$ *possède une* plus petite sorte *notée* $LS(t)$.

Le lemme qui suit (donné comme définition dans [GM87a]) permet de décider si une signature finie (i.e. ne comportant qu'un nombre fini de déclarations de sortes et de profils) est régulière. Nous avons préféré échanger définition et propriété caractéristique de l'article original de Goguen et Meseguer car la définition ci-dessus est bien plus explicite.

Lemme 4.35 *Une signature avec sortes ordonnées* Σ *est régulière ssi, pour tout* $f \in F$, *si*

- $f : \underline{s}_1 \times \ldots \times \underline{s}_n \to \underline{s}$
- $f : \underline{s}'_1 \times \ldots \times \underline{s}'_n \to \underline{s}'$
- *pour tout* i, $\underline{s}''_i \leq \underline{s}_i$ *et* $\underline{s}''_i \leq \underline{s}'_i$

alors il existe $\underline{s}^*_1, \ldots, \underline{s}^*_n, \underline{s}^*$ *tels que:*

- *pour tout* i, $\underline{s}''_i \leq \underline{s}^*_i \leq \underline{s}_i$ *et* $\underline{s}^*_i \leq \underline{s}'_i$
- $\underline{s}^* \leq \underline{s}$ *et* $\underline{s}^* \leq \underline{s}'$
- f *a le profil* $\underline{s}^*_1 \times \ldots \times \underline{s}^*_n \to \underline{s}^*$

Exemple 4.6
La signature de la figure 4.12 est régulière. Mais, si l'on remplace les profils de $+$:

$$\begin{array}{lcl} \underline{spos} \times \underline{pos} & \to & \underline{spos} \\ \underline{pos} \times \underline{spos} & \to & \underline{spos} \\ \underline{neg} \times \underline{sneg} & \to & \underline{sneg} \\ \underline{sneg} \times \underline{neg} & \to & \underline{sneg} \end{array}$$

par les profils:

$$\begin{array}{rcl} \underline{neg} \times \underline{neg} & \rightarrow & \underline{neg} \\ \underline{pos} \times \underline{pos} & \rightarrow & \underline{pos} \end{array}$$

la signature n'est plus régulière car $\underline{zero} \times \underline{zero} < \underline{pos} \times \underline{pos}$ et $\underline{zero} \times \underline{zero} < \underline{neg} \times \underline{neg}$ et f n'a pas de profil $\underline{zero} \times \underline{zero} \rightarrow \underline{s}$. De façon équivalente, il existe un terme $(0+0)$ qui n'a pas de plus petite sorte. $0+0$ est en effet de sorte $\underline{neg}$ et de sorte $\underline{pos}$ mais pas de sorte $\underline{zero}$.

Soit $\leftrightarrow_{\geq}$ la fermeture symétrique de $\geq$. La *composante connexe* d'une sorte $\underline{s} \in S$ est l'ensemble des sortes $\underline{s'} \in S$ telles que $\underline{s} \leftrightarrow_{\geq} \underline{s'}$. Pour assurer (entre autres) que $T(F)$ est une algèbre initiale on a besoin d'une propriété plus forte que la régularité:

Définition 4.36 *Une signature régulière* Σ *est* cohérente *si toute composante connexe de* S *a un élément maximal.*

Dans le cas de signatures cohérentes on note $top(\underline{s})$ la sorte maximale dans la composante connexe de $\underline{s}$. *Dans toute la suite nous ferons l'hypothèse que les signatures sont cohérentes et que la relation d'ordre* $\geq$ *sur les sortes est bien fondée.*

Soient Σ une signature avec sortes ordonnées et $\mathcal{A}$ une OSA. Une *$\mathcal{A}$-substitution* σ est une application de $T(\Sigma, X_0)$ dans $\mathcal{A}$, où X_0 est un sous-ensemble fini de X (le domaine de σ) telle que:

1. $\forall x \in X_0$, $x\sigma \in sort(x)_{\mathcal{A}}$
2. Si f a le profil $\underline{s}_1 \times \ldots \times \underline{s}_n \rightarrow \underline{s}$ et que, pour tout i, $t_i \in \underline{s}_{i,T(\Sigma,X_0)}$, alors $f(t_1, \ldots, t_n)\sigma =_{\mathcal{A}} f(t_1\sigma, \ldots, t_n\sigma)$.

La deuxième condition n'est autre que la compatibilité avec la structure d'OSA, la première condition permettant d'y donner un sens dans tous les cas.

La définition d'une solution d'un problème équationnel reste alors inchangée.

Comme dans le cas multi-sorte, lorsque $\mathcal{A} = T(\Sigma, X)$, les substitutions peuvent être prolongées en des endomorphismes de $T(F, X)$. La condition de régularité impose alors que, pour toute variable x, $sort(x) \geq LS(x\sigma)$.

Définition 4.37 *Une* équation *(resp. une* diséquation*) est une paire de termes (non orientée)* $s, t \in T(\Sigma, X)$ *telle que* $top(LS(s)) = top(LS(t))$.[9]

Remarquons que, si l'on choisit pour $\geq$ l'égalité, une signature multi-sorte est aussi une signature avec sortes ordonnées qui est cohérente. La définition ci-dessus généralise alors la définition précédente d'équation et de diséquation.

On définit la notion de théorie équationnelle définie par l'ensemble fini d'axiomes E comme dans le cas multi-sorte. Les définitions d'une formule équationnelle (ou d'un problème équationnel) sont aussi les mêmes que dans le cas multi-sorte; la seule différence étant la définition d'une équation. Par exemple, l'équation $x_1 : \underline{pos} = x_2 : \underline{neg}$ a un sens (bien que liant des termes de sortes différentes) et possède d'ailleurs pour unique solution

[9] Cette condition supplémentaire correspond à la condition d'égalité des sortes dans le cas multi-sorte.

dans $T(\Sigma)$ la substitution $\{x_1 \to 0;\ x_2 \to 0\}$. On voit ici pourquoi il faut autoriser les équations entre termes de sortes distinctes, mais on voit aussi que cela va entrainer des difficultés supplémentaires. Par exemple, le remplacement n'est plus correct car, dans un système $x = t \wedge u = v[x]$, il n'est pas certain que $v[t]$ soit un terme bien formé. Il nous faut aussi résoudre les problèmes comme celui de l'équation $x_1 = x_2$ ci-dessus.

4.4.2 OSA et automates d'arbres

Lors de la simplification des problèmes équationnels, il nous faudra résoudre des problèmes comme

$$\forall y : x : \underline{s} \neq y' : \underline{s'}$$

Un tel problème a pour solutions dans $T(F)$ les substitutions $\{x \to t\}$ où t est un terme quelconque de sorte $\underline{s}$ et pas de sorte $\underline{s'}$. Les formes résolues d'un tel problème donneront ainsi en particulier une description de $\underline{s}_{T(\Sigma)} - \underline{s'}_{T(\Sigma)}$.

Ce calcul du "complément" d'une sorte dans une autre n'est pas simple a priori. Mais si l'on considère les signatures avec sortes ordonnées comme des automates d'arbre, alors ce problème revient exactement au calcul d'un automate reconnaissant le complémentaire d'un langage régulier dans un autre. C'est la raison pour laquelle nous montrons tout d'abord qu'une signature avec sortes ordonnée n'est qu'un automate reconnaissant le langage des termes fermés bien formés ($T(F)$) et donc que nous pouvons appliquer aux signatures les transformations habituelles d'automates.

Définition 4.38 *Un* automate ascendant d'arbres *(aaa en abrégé) est un quadruplet* (A, Q, Q_f, R) *formé de:*

- *un ensemble de symboles de fonctions* A *("alphabet"), chaque symbole* $f \in A$ *étant muni d'une arité fixe* $|f| \in N$.
- *un ensemble* Q *d'états (symboles supposés disjoints de ceux de* A*)*
- *un ensemble d'états finaux* $Q_f \subseteq Q$
- *un ensemble de règles de transition* R *qui n'est autre qu'un système de réécriture* $\{u_i \to v_i\}_{i=1,\ldots,n}$ *où, pour tout* i, $u_i \in T(A, Q)$ *est un terme de profondeur au plus* 1[10] *et* $v_i \in Q$.

Cette définition n'est pas donnée sous sa forme classique comme dans [Dau84] mais peut être obtenue sans difficulté à partir de celle-ci.

Le langage reconnu par l'automate (A, Q, Q_f, R) est le sous-ensemble de $T(A)$ des termes t pour lesquels il existe un état final q tel que $t \to_R^* q$.

Un exemple fondamental pour notre problème est celui des signatures avec sortes ordonnées: si $(S, \geq, F)$ est une telle signature, posons $A = F$, $Q = S$, $Q_f = Q$. R est alors l'ensemble des règles

$$f(q_1, \ldots, q_n) \to q \qquad \text{et} \qquad q \to q'$$

pour tout profil $f : q_1 \times \ldots \times q_n \to q$ et toute déclaration de sous-sorte $q < q'$.

[10] Cette condition de profondeur n'est pas vraiment nécessaire mais il est toujours possible de s'y ramener en ajoutant des états intermédiaires (en nombre fini pour chaque membre gauche de règle).

$A = \{0, succ, pred, +\}$
$Q = \{q_0, q_p, q_n, q_{sp}, q_{sn}, q_{int}\}$, cahque état correspondant à une sorte.
$Q_f = \{int\}$
R est composé des règles correspondant aux déclarations de profils:

$$\begin{aligned}
0 &\rightarrow q_0 \\
succ(q_p) &\rightarrow q_{sp} \\
succ(q_{int}) &\rightarrow q_{int} \\
pred(q_n) &\rightarrow q_{sn} \\
pred(q_{int}) &\rightarrow q_{int} \\
q_p + q_{sp} &\rightarrow q_{sp} \\
q_{sp} + q_p &\rightarrow q_{sp} \\
q_n + q_{sn} &\rightarrow q_{sn} \\
q_{sn} + q_n &\rightarrow q_{sn} \\
q_{int} + q_{int} &\rightarrow q_{int}
\end{aligned}$$

et des règles d'inclusion de sortes:

$$\begin{aligned}
q_0 &\rightarrow q_p \\
q_0 &\rightarrow q_n \\
q_p &\rightarrow q_{int} \\
q_n &\rightarrow q_{int} \\
q_{sn} &\rightarrow q_n \\
q_{sp} &\rightarrow q_p
\end{aligned}$$

Figure 4.13: Exemple de signature décrite par un automate d'arbres

Lemme 4.39 *L'automate ainsi obtenu reconnait $T(\Sigma)$.*

Par exemple, à la signature avec sortes ordonnés de la figure 4.12 correspond l'automate de la figure 4.13.

Si l'on suppose que l'ensemble des sortes S de la signature est fini, on obtient un automate d'arbre régulier (d'états finis) et l'on peut utiliser les transformations classiques (qui conservent le langage reconnu, c'est-à-dire $T(\Sigma)$). Par exemple, il existe un automate déterministe équivalent. Ce qui signifie :

Proposition 4.40 *Pour toute signature avec sortes ordonnées finie Σ, il existe une signature finie et régulière Σ' telle que*

- $T(\Sigma) = T(\Sigma')$
- *l'ensemble des symboles de fonction de Σ' est identique à celui de Σ*
- *Si f a deux profils $w \rightarrow \underline{s}$ et $w' \rightarrow \underline{s}'$, alors, pour tout $w'' \in S'^*$, ou bien $w'' \not\leq w$ ou bien $w'' \not\leq w'$.*

On peut continuer ces transformations: la minimisation de l'automate entraine le résultat suivant:

Proposition 4.41 *Pour toute signature avec sortes ordonnées finie Σ, il existe une signature finie et régulière Σ' telle que*

- $T(\Sigma) = T(\Sigma')$
- *L'ensemble des symboles de fonction de Σ est identique à celui de Σ'*
- *pour toute sorte de Σ', $\underline{s}_{T(\Sigma)} \neq \emptyset$*
- $\Sigma' = (S', \geq', F)$ *et $\geq'$ est l'égalité (autrement dit, il n'y a pas d'inclusions de sortes)*

On est ainsi (presque) ramené au cas multi-sorte et cela sans ajouter de symbole de fonction. La seule différence est que les symboles de fonction peuvent éventuellement avoir plusieurs profils (mais ceux-ci sont alors "disjoints").

Nous appellerons signature *déterministe* (resp. *minimale*, resp. *ϵ-libre*) une signature dont l'automate associé a la propriété correspondante.

Si $\underline{s}$ est une sorte de Σ (ou un état de l'automate associé $\mathcal{A}$), nous noterons $\mathcal{L}(\underline{s}, \mathcal{A})$ l'ensemble $\underline{s}_{T(\Sigma)}$, qui est aussi le langage reconnu par l'automate $\mathcal{A}$ lorsqu'il est dans l'état $\underline{s}$.

Du point de vue de la résolution des problèmes équationnels, ces transformations de signature posent quand même certains problèmes. En effet, l'algèbre initiale $T(\Sigma)$ est bien conservée par ces transformations, mais nous ne savons rien des autres modèles. En particulier, les problèmes équationnels contiennent des occurrences de variables. Ces variables sont munies d'une sorte qui peut éventuellement ne plus apparaitre dans la nouvelle signature. Plus précisément, il faudrait que, pour toute sorte $\underline{s}$ de Σ, il existe un ensemble fini $\{\underline{s}_1, \ldots, \underline{s}_n\}$ de Σ' tel que $\underline{s}_{T(\Sigma)} = \bigcup_{i=1,\ldots,n} \underline{s}_{i,T(\Sigma)}$. De cette façon il serait possible de transformer un problème $\mathcal{P}[x : \underline{s}]$ en $\mathcal{P}[x_1 : \underline{s}_1] \vee \ldots \vee \mathcal{P}[x_n : \underline{s}_n]$, et exprimer ainsi tout problème construit sur Σ en un problème (ou un ensemble fini de problèmes) "équivalent" et construit sur Σ'. Nous pourrions alors faire les hypothèses de déterminisme ou de minimalité sans perdre de généralité.

Cette propriété de la signature Σ' peut effectivement être assurée sans trop de problème:

Proposition 4.42 *Soit $\Sigma = (S, \geq, F)$ une signature finie avec sortes ordonnées. Il est possible de calculer une signature finie régulière $\Sigma' = (S', \geq', F')$ telle que*

1. $T(\Sigma) = T(\Sigma')$
2. *Σ' est déterministe (et par suite, si $\underline{s}_1, \underline{s}_2 \in \Sigma'$, alors $\underline{s}_{1,T(\Sigma)} \cap \underline{s}_{2,T(\Sigma)} = \emptyset$)*
3. *pour tout sorte $\underline{s} \in S$, il existe un ensemble (fini) $\mathcal{D}(\underline{s}, \Sigma, \Sigma')$ tel que*

$$\underline{s}_{T(\Sigma)} = \bigcup_{\underline{s}' \in \mathcal{D}(\underline{s}, \Sigma, \Sigma')} \underline{s}'_{T(\Sigma)}$$

Preuve

Il suffit de montrer que la propriété 3 est vérifié à chaque étape de la déterminisation de l'automate. Cette déterminisation s'effectue classiquement en trois étapes : 1) élimination des ϵ-transitions 2) "complétion" de l'automate pour obtenir un automate complètement spécifié 3) déterminisation proprement dite en passant à l'ensemble des parties. On peut aussi y ajouter l'élimination des états inaccessibles. Cette étape comme celle de "complétion" (qui consiste essentiellement à ajouter un état puits) n'interviennent pas dans notre problème. Le résultat de la proposition est alors une conséquence des remarques suivantes:

1. Si $\mathcal{A} = (A, Q, Q_f, R)$ est transformé en $\mathcal{A}' = (A, Q', Q'_f, R')$ par élimination des ϵ-transitions ($Q' \subseteq Q$), alors, pour tout $q \in Q$,

$$\mathcal{L}(q, \mathcal{A}) = \mathcal{L}(q, \mathcal{A}') \bigcup_{q \to_R^+ q'} \mathcal{L}(q', \mathcal{A})$$

   et, la relation $\to_R$ étant à terminaison finie, on peut bien obtenir la relation souhaitée.

2. Si $\mathcal{A} = (A, Q, Q_f, R)$ est transformé en $\mathcal{A}' = (A, Q', Q'_f, R')$ par déterminisation proprement dite ($Q' \subseteq \mathcal{P}(Q)$), alors, pour tout $q \in Q$,

$$\mathcal{L}(q, \mathcal{A}) = \bigcup_{q \in q' \in Q'} \mathcal{L}(q', \mathcal{A}')$$

□

Corollaire 4.43 *Soit $\Sigma = (S, \geq, F)$ une signature finie avec sortes ordonnées. Il est possible de calculer une signature finie régulière $\Sigma' = (S', \geq', F')$ telle que*

1. $T(\Sigma) = T(\Sigma')$
2. $S \subseteq S'$
3. *pour tout sorte $\underline{s} \in S$, il existe un ensemble (fini) $\mathcal{D}(\underline{s}, \Sigma, \Sigma')$ tel que*

$$\underline{s}_{T(\Sigma)} = \bigcup_{\underline{s}' \in \mathcal{D}(\underline{s}, \Sigma, \Sigma')} \underline{s}'_{T(\Sigma)}$$

4. *pour toutes sortes $\underline{s}, \underline{s}' \in S'$, $\underline{s} \geq' \underline{s}'$ ssi*
   - *ou bien $\underline{s}, \underline{s}' \in S$ et $\underline{s} \geq \underline{s}'$*
   - *ou bien $\underline{s} \in S$ et $\underline{s}' \in \mathcal{D}(\underline{s}, \Sigma, \Sigma')$*
5. *pour toutes sortes $\underline{s}, \underline{s}' \in S' - S$, $\underline{s}_{T(\Sigma)} \cap \underline{s}'_{T(\Sigma)} = \emptyset$.*

Une telle signature (Σ') sera dite *complète*. On parlera aussi de la *complétée de* Σ.

La transformation des problèmes équationnels évoquée ci-dessus peut se révéler inutile et coûteuse. Il est donc préférable de ne pas l'effectuer systématiquement et de ne faire intervenir la signature complétée de Σ que lorsque c'est nécessaire. Typiquement dans le cas d'une diséquation $x \neq y$ où y est un paramètre, x une variable et $sort(x) > sort(y)$.

4.4.3 Transformation des problèmes équationnels

Nous supposerons dans toute cette section que la signature est complète. (Obtenue le cas échéant à partir d'une signature qui ne l'est pas). On dira alors que deux sortes sont *disjointes* si elles n'ont pas de sous-sorte commune. Vu les propriétés des signatures complètes, $\underline{s}$ et $\underline{s'}$ sont disjointes ssi $\underline{s}_{T(\Sigma)} \cap \underline{s'}_{T(\Sigma)} = \emptyset$.

Toutes les règles des figures 3.1, 3.2, 3.3 sont correctes et adéquates dans $T(\Sigma)$, excepté les règles $(R_1), (R_2), (EP_2)$ qui pourraient conduire par remplacement à des termes mal formés. (Plus simplement, les substitutions apparaissant dans ces règles ne sont plus nécessairement des substitutions dans une OSA). On restreint donc l'emploi de ces règles aux cas où les substitutions appliquées sont de la forme $\{x \rightarrow t\}$ *avec* $sort(x) \geq LS(t)$.

Malheureusement, ces restrictions invalident la propriété de complétude: il faut ajouter des règles permettant de traiter les cas $x = t$ et $y \neq t$ lorsque $sort(x) \not\geq LS(t)$ ou $sort(y) \not\geq LS(t)$.

La résolution des équations $x = t$ est déjà connue (voir par exemple [Kir88]), celles de résolution de $y \neq t$ sont obtenues (à peu de choses près) par négation des premières. Il faut y ajouter les règles de "complément de sorte" déjà évoquées dans le paragraphe précédent. Toutes ces règles sont données dans les figures 4.14 et 4.15.[11]

Certains des membres droits des règles proposées dans ces figures ne sont pas en forme normale conjonctive. Pour des raisons de simplicité d'écriture, nous ne donnons pas les règles obtenues par normalisation des membres droits.

Sonnons un exemple de transformation avant de nous intéresser aux résultats de correction et de complétude des règles.

Exemple 4.7
Nous reprenons la signature de la figure 4.12 qui vérifie les hypothèses requises pour la transformation puisque, (dans $T(\Sigma)$) chaque sorte est la réunion de ses sous-sortes.

x_1 et x_2 sont des inconnues de sorte $\underline{pos}$. On considère le problème de complément:

$$\mathcal{P} \equiv \forall y_1, y_2 : \underline{pos} \; : \; x_1 + x_2 \neq 0 + y_1 \wedge x_1 + x_2 \neq s(y_1) + y_2 \wedge x_1 + x_2 \neq p(y_1) + y_2$$

Il est tout d'abord possible d'appliquer les décompositions (D_2). On obtient alors le problème:

$$\forall y_1, y_2 : (x_1 \neq 0 \vee x_2 \neq y_2) \wedge (x_1 \neq s(y_1) \vee x_1 \neq y_2) \wedge (x_1 \neq p(y_1) \vee x_2 \neq y_2)$$

Comme $sort(y_2) \geq sort(x_2)$ il est possible d'appliquer les règles (EP_0, EP_2) pour obtenir le système:

$$\forall y_1 : x_1 \neq 0 \wedge x_1 \neq s(y_1) \wedge x_1 \neq p(y_1)$$

Il faut maintenant appliquer la règle d'explosion qui conduit à quatre problèmes équationnels. Mentionnons en un:

$$\exists w_1 : \underline{int}, \forall y_1 : x_1 = s(w_1) \wedge x_1 \neq 0 \wedge x_1 \neq s(y_1) \wedge x_1 \neq p(y_1)$$

[11] Par convention, une conjonction sur un ensemble vide a pour résultat $\top$.

Résolution des équations $x = t$ (ER)

$$(ER_1) \quad \mathcal{P}[z_1 : \underline{s}_1 = z_2 : \underline{s}_2] \quad \mapsto \quad \exists w_3 : \underline{s}_3, \mathcal{P}[z_1 = w_3 \wedge z_2 = w_3]$$

Si

1. z_1 et z_2 sont des inconnues
2. $w_3 \notin Var(\mathcal{P})$
3. $\underline{s}_1 \bowtie \underline{s}_2$
4. $\underline{s}_3 \in \max\{\underline{s} \in S | \underline{s} \leq \underline{s}_1 \ \& \ \underline{s} \leq \underline{s}_2\}$

$$(ER_2) \quad \mathcal{P}[z = f(t_1, \ldots, t_n)] \quad \mapsto \quad \exists w_1, \ldots, w_n, \mathcal{P}[z = f(w_1, \ldots, w_n) \wedge w_1 = t_1 \wedge \ldots \wedge w_n = t_n]$$

Si

1. z est une inconnue
2. $LS(f(t_1, \ldots, t_n)) \not\leq sort(z)$
3. pour tout i, $sort(w_i) = \underline{s}_i$ vérifient:

$$\underline{s}_1 \ldots \underline{s}_n \in \max\{\underline{s}'_1 \ldots \underline{s}'_n \mid f : \underline{s}'_1 \times \ldots \times \underline{s}'_n \rightarrow \underline{s} \ \text{ et } \ \underline{s} \leq sort(z)\}$$

4. $t_1, \ldots, t_n$ ne contiennent pas de paramètre

$$(ER_3) \quad z_1 : \underline{s}_1 = z_2 : \underline{s}_2 \quad \mapsto \quad \bot$$

Si z_1 et z_2 sont des inconnues et qu'il n'existe aucune sorte $\underline{s}$ telle que $\underline{s} \leq \underline{s}_1$ et $\underline{s} \leq \underline{s}_2$.

Figure 4.14: Règles de résolution de $x = t$ dans les OSA

Elimination des paramètres dans les diséquations

$$(EP_2') \quad \forall \vec{y} : P \wedge (y_1 \neq y_2 \vee d) \mapsto \forall \vec{y}, \vec{y'} : P \bigwedge_{sort(y_3) \in \max\{\underline{s} \in S \mid \underline{s} \leq sort(y_1) \& \underline{s} \leq sort(y_2)\}} d\{y_1 \rightarrow y_3;\ y_2 \rightarrow y_3\}$$

Si

1. y_1 et y_2 sont des paramètres tels que $sort(y_1) \bowtie sort(y_2)$
2. $y_3 \notin Var(P, y_1, y_2, d)$

$$(EP_2'') \quad \forall \vec{y} : P \wedge (y \neq f(t_1, \ldots, t_n) \vee d) \mapsto \forall \vec{y}, \vec{y'} : P \bigwedge_{\phi} (y_1 \neq t_1 \vee \ldots \vee y_n \neq t_n \vee d\{y \rightarrow f(y_1, \ldots, y_n)\})$$

Si

1. ϕ désigne $f : \underline{s}_1 \times \ldots \times \underline{s}_n \rightarrow \underline{s}, \underline{s} \leq sort(y), \underline{s}_1 \ldots \underline{s}_n$ maximal
2. $y \in \vec{y}$ et $y_1, \ldots, y_n \in \vec{y'}$
3. $\vec{y} \cap \vec{y'} = \emptyset$
4. $sort(y) \not\geq LS(f(t_1, \ldots, t_n))$
5. $y_1, \ldots, y_n$ sont distincts et, pour tout i, $sort(y_i) = \underline{s}_i$.

Règles de "compléments de sortes" (CS)

$$(CS_1) \quad \mathcal{P}[y : \underline{s} \neq z : \underline{s'}] \mapsto \exists w : \underline{s''}, \mathcal{P}[y : \underline{s} \neq w : \underline{s''} \wedge z : \underline{s'} = w : \underline{s''}]$$

Si

1. Si y est un paramètre et z est une inconnue
2. $\underline{s'} \not\leq \underline{s}$
3. $\underline{s''} \in \max\{\underline{s}_1 \in S | \underline{s}_1 < \underline{s'}\}$

$$(CS_2) \quad y : \underline{s} \neq z : \underline{s'} \mapsto \top$$

Si y est un paramètre, z est une variable, et $\underline{s}$ et $\underline{s'}$ sont deux sortes disjointes.

Figure 4.15: Elimination des paramètres dans les OSA : règles complémentaires

Notons qu'ici le remplacement ne peut être appliqué. Mais, par fusion, décomposition et incompatibilité, on obtient:

$$\exists w_1, \forall y_1 : x_1 = s(w_1) \wedge y_1 \neq w_1$$

Comme $sort(y_1) \not\geq sort(w_1)$, il n'est pas possible d'appliquer la règle (EP_2) et il faut appliquer la règle de complément de sortes (CS_1) pour obtenir les deux problèmes:

$$\mathcal{P}_1 \equiv \exists w_1 : \underline{int}, w_2 : \underline{pos}, \forall y_1 : \underline{pos} : x_1 = s(w_1) \wedge w_1 = w_2 \wedge w_2 \neq y_1$$

et

$$\mathcal{P}_2 \equiv \exists w_1 : \underline{int}, w_2 : \underline{neg}, \forall y_1 : \underline{pos} : x_1 = s(w_1) \wedge w_1 = w_2 \wedge w_2 \neq y_1$$

Il n'est pas nécessaire ici d'introduire de nouvelles sortes puisque tout terme fermé est soit de sorte $\underline{pos}$ soit de sorte $\underline{neg}$ comme l'automate minimal le montre. Il est cette fois possible d'appliquer l'élimination des paramètres à $\mathcal{P}_1$ puisque $sort(y_1) \geq sort(w_2)$. On obtient alors $\perp$. Intéressons nous donc désormais à $\mathcal{P}_2$. Par une nouvelle application de (CS_1) on obtient les deux problèmes:

$$\mathcal{P}_3 \equiv \exists w_1 : \underline{int}, w_2 : \underline{neg}, w_3 : \underline{zero}, \forall y_1 : \underline{pos} : x_1 = s(w_1) \wedge w_1 = w_2 \wedge w_2 = w_3 \wedge w_3 \neq y_1$$
$$\mathcal{P}_4 \equiv \exists w_1 : \underline{int}, w_2 : \underline{neg}, w_3 : \underline{sneg}, \forall y_1 : \underline{pos} : x_1 = s(w_1) \wedge w_1 = w_2 \wedge w_2 = w_3 \wedge w_3 \neq y_1$$

$\mathcal{P}_3$ se transforme à nouveau en $\perp$ par application de (EP_2). Il ne reste plus alors que $\mathcal{P}_4$. Par la règle (CS_2) le problème est transformé en:

$$\exists w_1, w_2, w_3, \forall y_1 : x_1 = s(w_1) \wedge w_1 = w_2 \wedge w_2 = w_3$$

Il est alors possible d'appliquer les règles de remplacement et de nettoyage et l'on obtient:

$$\exists w_3 : \underline{sneg} : x_1 = s(w_3)$$

Par (ER_2) on obtient ensuite:

$$\exists w_3 : \underline{sneg}, w_4 : \underline{pos} : x_1 = s(w_4) \wedge w_4 = w_3$$

Mais les sortes de w_4 et w_3 ne possèdent pas de sous-sorte commune et, par la règle ER_3 on obtient le problème $\perp$.

Proposition 4.44 *Supposons que $\mathcal{A} = T(\Sigma)$. Les règles des figures 4.14 et 4.15 sont $\mathcal{A}$ adéquates.*

Preuve

L'adéquation des règles de la figure 4.14 et la forte adéquation des règles $(EP_2'), (EP_2'')$ se déduisent l'une de l'autre par complémenataire. Nous ne considérerons donc que les règles $(ER_1), (ER_2), (CS_1)$ et (CS_2). Remarquons de plus que la correction des règles (ER_1), (ER_2), (ER_3) et (CS_1) est triviale.

(ER_1) est globalement conservative

Si σ valide $z_1 : \underline{s}_1 = z_2 : \underline{s}_2$, alors $z_1\sigma$ et $z_2\sigma$ ont même sorte. Il existe donc une sorte $\underline{s} \leq \underline{s}_1, \underline{s} \leq \underline{s}_2$ telle que $z_1\sigma \equiv z_2\sigma : \underline{s}$. Alors, pour $\underline{s}' \in \max\{\underline{s}'' \in S \mid \underline{s}'' \leq \underline{s}_1, \underline{s}'' \leq \underline{s}_2, \underline{s}'' \geq \underline{s}\}$ (qui existe par régularité) et ρ de domaine $w_3 : \underline{s}'$ telle que $w_3\rho \equiv z_2\sigma$, $\sigma\rho$ valide $z_1 = w_3 \wedge z_2 = w_3$. D'où le résultat.

(ER_2) est globalement conservarive
Ce résultat s'obtient par un raisonnement analogue

(ER_3) est fortement adéquate
Aucune substitution ne vérifie en effet $z_1\sigma \equiv z_2\sigma$ puisque les sortes de $z_1\sigma$ et de $z_2\sigma$ ne peuvent être identiques.

(CS_1) est globalement conservative
Il suffit de remarquer que, si une sorte $\underline{s}$ n'est pas minimale, alors $\underline{s}_{T(\Sigma)} = \bigcup_{\underline{s}'<\underline{s}} \underline{s}'_{T(\Sigma)}$ lorsque la signature Σ est complète.

(CS_2) est correcte
Les sortes étant disjointes, pour toute $T(\Sigma)$-substitution σ $LS(y\sigma) \neq LS(z\sigma)$ et donc $y\sigma \not\equiv z\sigma$.

□

Il reste maintenant à donner un résultat analogue à celui de la section 3.6. Malheureusement, parmi les nouvelles règles il y en a (dans la figure 4.15 notamment) qui introduisent de nouveaux paramètres et il y en a qui introduisent de nouvelles variables auxiliaires.

Considérons tout d'abord le système $\mathcal{R}_e$ formé des seules règles (EP_2), (EP'_2), (EP''_2), (CS_1), (CS_2) avec le contrôle suivant : on applique en priorité (EP'_2), (EP''_2), (CS_1), (CS_2) (qui constituent le système $\mathcal{R}_{e_1}$) tant que c'est possible, puis (EP_2) seule tant que c'est possible. Soit $\mathcal{P} \equiv \exists\vec{w},\forall\vec{y} : d_1 \wedge \ldots \wedge d_n$. On note alors $\phi_1(\mathcal{P})$ le multi-ensemble $\{k_1,\ldots,k_n\}$ où k_i est le nombre de paramètres distincts dans d_i et $\phi_2(\mathcal{P})$ le multi-ensemble $\{a_1,\ldots,a_n\}$ où a_i est le multi-ensemble des sortes des paramètres ayant une occurrence dans d_i. On obtient alors le résultat suivant:

Lemme 4.45 *$\mathcal{R}_e$ est à terminaison finie et, si $\mathcal{P}'$ est une forme irréductible de $\mathcal{P}$ par $\mathcal{R}_e$ telle que $\mathcal{P} \not\equiv \mathcal{P}'$, alors -ou bien $\phi_1(\mathcal{P}') < \phi_1(\mathcal{P})$ -ou bien $\phi_1(\mathcal{P}') = \phi_1(\mathcal{P})$ et $\phi_2(\mathcal{P}') < \phi_2(\mathcal{P})$.*

Preuve
Il suffit de remarquer que la fonction d'interprétation (ϕ'_1,ϕ_2,ϕ_3) où $\phi'_1(\mathcal{P})$ est le multi-ensemble des nombres de paramètres non presque résolus dans chaque disjonction, $\phi_3(\exists\vec{w},\forall\vec{y} : d_1 \wedge \ldots \wedge d_n)$ est le multiensemble $\{\psi_3(d_1),\ldots,\psi_3(d_n)\}$ et $\psi_3(d)$ est le multi-ensemble des sortes des inconnues z apparaissant dans une diséquation $y \neq z$ de d, y étant un paramètre, est strictement décroissante par application d'une règle de $\mathcal{R}_{e_1}$. Rappelons qu'un paramètre est presque résolu dans une disjonction d'équations et de diséquations d s'il a une occurrence comme membre d'une diséquation de d. Alors, après application répétée de (EP_2), les paramètres presque résolus disparaissent. D'où le résultat. □

On procède de la même façon pour la règle (R_1); soit $\mathcal{R}_{r_1}$ le système formé des règle **ER** et soir $\mathcal{R}_r$ le système constitué des règles **ER** et (R_1) avec le contrôle suivant: on applique en priorité **ER** tant que c'est possible, puis (R_1) tant que c'est possible. Soit $\mathcal{P}$ un problème équationnel, $\phi_4(\mathcal{P})$ le nombre d'inconnues de $\mathcal{P}$ qui ne sont pas résolues et $\phi_5(\mathcal{P})$ le multi-ensemble des sortes des inconnues de $\mathcal{P}$. Alors

Lemme 4.46 *$\mathcal{R}_r$ est à terminaison finie et, si $\mathcal{P}'$ est une forme irréductible de $\mathcal{P}$ par $\mathcal{R}_r$ telle que $\mathcal{P} \not\equiv \mathcal{P}'$, alors - ou bien $\phi_4(\mathcal{P}') < \phi_4(\mathcal{P})$ -ou bien $\phi_4(\mathcal{P}') = \phi_4(\mathcal{P})$ et $\phi_5(\mathcal{P}') < \phi_5(\mathcal{P})$.*

Preuve
Elle est analogue à celle du lemme 4.45. □

Il ne reste plus que la règle (R_2) que nous n'avons pas remplacée. mais nous ne nous en préoccuperons pas car elle n'est pas présente dans le système de la section 3.6.

Il suffit maintenant de substituer à la règle (EP_2) de la section 3.6 la réduction par $\mathcal{R}_e$ et de substituer à la règle (Re_1) la réduction par $\mathcal{R}_r$. Soit $\mathcal{R}_{OSA}$ le système ainsi obtenu.

Définition 4.47 *Un problème est dit être une* définition contrainte *s'il est égal à $\bot$, $\top$ ou bien est de la forme*

$$\mathcal{P} \equiv \exists \vec{w} : x_1 = t_1 \wedge \ldots \wedge x_n = t_n \wedge x'_1 \neq t'_1 \wedge \ldots \wedge x'_m \neq t'_m$$

avec les propriétés:

- *$x_1, \ldots, x_n, x'_1, \ldots, x'_n$ sont des inconnues*
- *$x_1, \ldots, x_n$ n'ont qu'une occurrence dans $\mathcal{P}$*
- *pour tout i, t_i est un terme distinct de x_i et tel que $sort(x_i) \geq LS(t_i)$*
- *pour tout j, x'_j est distinct de t'_j*

Théoreme 4.48 *$\mathcal{R}_{OSA}$ est à terminaison finie et les formes irréductibles pour $\mathcal{R}_{OSA}$ sont des définitions contraintes.*

Preuve
La terminaison est une conséquence des lemmes précédents et des résultats du chapitre 3. Les formes irréductibles sont des définitions contraintes, comme dans le chapitre 3: il suffit de vérifier que tous les problèmes qui ne sont pas des définitions contraintes sont bien réductibles. Cette vérification est laissée au lecteur.

Les définitions contraintes possèdent, comme dans le cas multi-sorte, la bonne propriété d'être solubles, c'est-à-dire de posséder au moins une solution dans $T(\Sigma)$.

Enfin, un corollaire de complétude peut être déduit de ce théorème, comme dans tous les cas précédents. En conclusion, nous pouvons dire que tous les résultats du cas multi-sorte se généralisent au cas "sortes ordonnées".

4.5 Résolution progressive

dans cette section, nous modifions légèrement la syntaxe des problèmes équationnels en distinguant, à l'aide des symboles $\approx$ et $\not\approx$ les équations et diséquations "déjà résolues" des équations et diséquations "à résoudre"/ Cette distinction présente au moins trois avantages:

1. Cela permet d'exprimer plus facilement certains contrôles (par exemple exploiter un ordre partiel sur les variables comme dans [MM82]).
2. Cela permet certaines optimisations dans la recherche des règles applicables. Par exemple, aucune règle ne modifiant les équations résolues, il est inutile d'en tenir compte dans la recherche de l'applicabilité d'une règle.
3. Nous pourrons aller plus loin dans la simplification des problèmes équationnels, en éliminant les "cycles dans les diséquations".

Par exemple, le système:

$$z_1 \neq f(z_2) \wedge z_2 \neq f(z_1)$$

peut être considéré comme comportant un cycle. La raison essentielle de l'élimination de tels cycles est la construction des solutions d'un problème en définition contrainte. Nous savons qu'un tel problème est soluble, mais n'avons pas d'algorithme simple permettant d'en énumérer les solutions. Or il est possible de transformer le problème ci-dessus en

- $z_1 = 0 \wedge z_2 \neq f(0)$
- $\exists w : z_1 = f(w) \wedge z_2 \neq w \wedge z_2 \neq f(f(w))$

qui ne comportent pas de cycles et pour lesquels on peut considérer qu'il est plus aisé d'énumérer les solutions: il suffit de donner une valeur arbitraire à w puis de choisir pour z_2 des valeurs différentes de w et de $f(f(w))$. Dans la première formulation, il aurait fallu effectuer un étape de simplification après avoir substitué x par une valeur (étape de simplification qui peut d'ailleurs conduire à un échec). Enfin, il apparait clairement dans cette dernière formulation que, si $A \subseteq \mathcal{A}$ est de cardinal au moins n, alors il y a au moins $n - 2$ solutions dans A pour y.

Une telle transformation ne peut s'effectuer avec les outils dont nous disposons jusqu'à présent sans un contrôle très complexe. Prenons en effet l'exemple :

$$z_1 \neq f(f(z_2)) \wedge z_2 \neq f(z_1)$$

Par explosion de z_1 on obtient (entre autres)

$$\exists w : z_1 = f(w) \wedge w \neq f(z_2) \wedge z_2 \neq f(f(w))$$

qui contient, à renommage près, le problème initial. Il ne fallait en effet pas appliquer l'explosion à z_1 mais à z_2. Dans le cas général, il ne semble pas y avoir d'argument simple permettant de décider à quelle variable il faut appliquer l'explosion. Avec l'introduction des nouveaux symboles $\approx$ et $\not\approx$, nous pourrons facilement résoudre ce problème: Choisissons une variable quelconque. Dans l'exemple précédent supposons que nous ayons fait le mauvais choix, c'est-à-dire z_2. Le système initial est transformé en:

$$z_1 \neq f(f(z_2)) \wedge z_2 \not\approx f(z_1)$$

brisant ainsi la symétrie. Le symbole $\not\approx$ nous interdit alors d'effectuer l'explosion sur la variable qui en est membre gauche. Comme ci-dessus, on obtient par explosion le problème:

$$\exists w : z_1 = f(w) \wedge w \neq f(z_2) \wedge z_2 \not\approx f(f(w))$$

$$(P_1) \quad \mathcal{P}[x = t] \quad \mapsto \quad \mathcal{P}[x \approx t]$$

Si x est une inconnue principale et x n'a qu'une occurrence dans $\mathcal{P}$.

$$(P_2) \quad \mathcal{P}[z \neq t] \quad \mapsto \quad \mathcal{P}[z \not\approx t]$$

Si z est une inconnue, $z \notin Var(t)$, t ne contient pas de variable qui soit membre gauche d'une équation résolue et aucune autre règle ne s'aoolique.

Figure 4.16: Transformations faisant intervenir des équations ou diséquations résolues

mais ce problème ne contient pas le problème de départ au renommage près des variables. Nous pouvons maintenant seulement appliquer l'explosion à w; on obtient:

$$\exists w, w' : z_1 = f(f(w')) \wedge w = f(w') \wedge z_2 \not\approx w' \wedge z_2 \not\approx f(f(f(w')))$$

qui ne contient pas de cycle dans les diséquations.

4.5.1 Introduction des signes $\approx$ et $\not\approx$

La syntaxe d'un problème équationnel est modifiée de la façon suivante:

- Une *équation résolue* est une paire orientée (x, t) de termes de même sorte notée $x \approx t$ où x est une variable et t est un terme où n'apparait pas x.
- Une *équation à résoudre* est une paire *non orientée* de termes u, v de même sorte notée $u = v$.
- une *équation* est soit une équation résolue soit une équation à résoudre
- une *diséquation résolue* est une paire *orientée* (z, v) de termes de même sorte notée $z \not\approx v$ où z est une variable.
- Une diséquation *à résoudre* est une paire *non orientée* u, v de termes de même sorte notée $u \neq v$.
- une *diséquation* est une diséquation résolue ou bien une diséquation à résoudre.

Toutes les autres définitions du chapitre 2 restent inchangées. En particulier, une solution dans $\mathcal{A}$ de $u \approx v$ est une $\mathcal{A}$-substitution σ telle que $u\sigma =_{\mathcal{A}} v\sigma$. En particulier, les règles de transformation ne faisant intervenir que des équations et diséquations à résoudre sont les mêmes que celles du chapitre 3. Il nous faut seulement ajouter les règles relatives aux équations et diséquations résolues. Celles-ci sont données dans la figure 4.16.

Nous utilisons aussi les règles du chapitre 3, avec un contrôle qui est, à peu de choses près celui de la section 4.1. Les règles du système $\mathcal{R}_p$ ainsi obtenu sont donc celle du

$$(Ex_p) \quad \mathcal{P}[z \neq t[z']] \quad \mapsto \quad \exists w_1, \ldots, w_n, \mathcal{P} \wedge z = f(w_1, \ldots, w_n)$$

Si

1. $w_1, \ldots, w_n \notin Var(\mathcal{P})$
2. $f \in F$
3. Il existe dans $\mathcal{P}$ une diséquation résolue dont le membre gauche est z'
4. Il existe dans $\mathcal{P}$ une diséquation à résoudre dont un des membres est z et l'autre un terme non variable contenant z
5. La seule autre règle éventuellement applicable est la règle (P_2)
6. Le problème ne contient pas de paramètre

Figure 4.17: Règle d'explosion dans le cas de la résolution progressive

système $\mathcal{R}_\ni$ (moins l'explosion) plus les règles $(P_1), (P_2), (Nc), (Ex_p)$, cette derniere étant donnée dans la figure 4.17[12]. Nous supposons de plus (par souci de simplicité) que les problèmes considérés ne contiennent ni paramètre, ni équation résolue, ni diséquation résolue : $\mathcal{F}_I$ est l'ensemble des problèmes sans paramèrte tel qu'il a été défini dans le chapitre 3.

Les formes résolues sont alors de la forme:

$$\exists \vec{w} : x_1 \approx t_1 \wedge \ldots \wedge x_n \approx t_n \wedge w_1 \not\approx u_1 \wedge \ldots \wedge w_m \not\approx u_m$$

Avec les propriétés habituelles des définitions contraintes:

- $x_1, \ldots, x_n$ sont des inconnues n'ayant qu'une occurrence dans le problème
- Pour tout i, w_i est une inconnue et $w_i \notin Var(u_i)$
- Toute variable qui apparait dans une diséquation est infinitaire

Mais aussi les propriétés dues à la "mise en forme" des problèmes:

- $\mathcal{I} = \{x_1, \ldots, x_n\}$
- $Var(w_1, \ldots, w_n, u_1, \ldots, u_n) \subseteq Var(t_1, \ldots, t_n)$

Et enfin les propriétés particulières à la résolution progressive:

- pour tous i, j distincts, $w_i \notin Var(u_j)$

[12]Il nous a paru inutile de rappeler ici l'ensemble de toutes les règles. De même, nous n'allons pas développer toutes les preuves comme dans le chapitre 3.

Cette dernière propriété exprimant "l'absence de cycle" dans les diséquations.

Appelons *problèmes sans cycles* les formes résolues ainsi définies. Nous pouvons alors énoncer comme prévu le résultat de terminaison et de complétude correspondant:

Théoreme 4.49 *$\mathcal{R}_p$ est à terminaison finie et ses formes irréductibles sont des problèmes sans cycle.*

Preuve

D'après les résultats des sections précédentes, il ne peut y avoir de chaine infinie de transformations qui ne fasse intervenir une des règles $(P_1), (P_2), (Ex_p)$. Le nombre d'applications possibles de (P_1) est par ailleurs limité au nombre d'inconnes de $\mathcal{I}$. Nous sommes donc ramenés à prouver la terminaison de la combinaison de $(P_2), (Ex_p)$ et de la réduction par les autres règles, soit $\mathcal{R}_{p_1}$. Le contrôle imposant en priorité une réduction par $\mathcal{R}_{p_1}$, ensuite une réduction par (Ex_p) et enfin une application de (P_2).

Soit $\mathcal{P} \mapsto \ldots \mapsto \mathcal{P}_i \mapsto \ldots$ une chaine de transformations. Comme (P_1) n'est appliquée qu'un nombre fini de fois dans cette chaine, il existe un indice i_0 tel que, pour $j > i_0$, (P_1) n'est pas appliquée à $\mathcal{P}_j$. Soit alors $\mathcal{V}$ l'ensemble des variables apparaissant dans les membres droits d'équations de $\mathcal{P}_{i_0+1}$. Pour tout $j > i_0$, $\mathcal{V}$ est l'ensemble des variables des membres droits des équations résolues de $\mathcal{P}_j$.

Si $j > i_0$ et $\mathcal{P}_j \mapsto_{P_2} \mathcal{P}_{j+1}$, comme aucune règle de $\mathcal{R}_{p_1}$ n'est applicable à $\mathcal{P}_j$, ce problème ne contient pas d'équations à résoudre. En effet celles-ci pourraient être réduites par une règle de mise en forme. De plus, aucune diséquation de $\mathcal{P}_j$ ne peut être éliminée par une règle de mise en forme, ce qui signifie que Les variables des diséquations de $\mathcal{P}_j$ sont des variables de $\mathcal{V}$. Ainsi l'ensemble des variables qui sont membres gauches d'une diséquation résolue reste toujours contenu dans l'ensemble fini $\mathcal{V}$.

Notons alors $\phi_(j)$ le nombre de disjonctions de une ou plusieurs diséquations dans $\mathcal{P}_j$, $\phi_2(j)$ le nombre de variables de $\mathcal{V}$ qui sont membre gauche d'une diséquation de $\mathcal{P}_j$ et $\phi_3(j)$ le nombre d'occurrence d'une variable de $\mathcal{V}$ dans les diséquations non résolues de $\mathcal{P}_j$. Si $\mathcal{P}_{j_1} \mapsto_{P_2} \mathcal{P}_{j_1+1}$ et $\mathcal{P}_{j_2} \mapsto_{P_2} \mathcal{P}_{j_2+1}$, avec $j_2 > j_1 > i_0$, alors $(\phi_1(j_2), \phi_2(j_2), \phi_3(j_2)) < (\phi_1(j_1), \phi_2(j_1), \phi_3(j_2))$. ϕ_1 est en effet toujours décroissante au sens large car $\mathcal{P}_{j_1}$ ne peut contenir d'équation non résolue. (ϕ_1 ne décroit strictement que par "effet de bord" d'une transformation remplaant une diséquation par $\top$ ou $\bot$).Si ϕ_1 n'a pas été modifiée, les variables qui sont membre gauche d'une diséquation résolue de $\mathcal{P}_{j_1}$ sont aussi membre gauche d'une diséquation résole de $\mathcal{P}_{j_2}$. ϕ_2 est donc décroissante au sens large. Mais, si ϕ_1 et ϕ_2 sont constantes, le nombre d'occurrence des variables de $\mathcal{V}$ est resté constant, et ϕ_3 a par conséquent décrû. Cela prouve que (P_2) ne peut être appliquée qu'un nombre fini de fois.

Il reste seulement à prouver qu'il ne peut y avoir une infinité d'applications de (Ex_p) pour achever la preuve de terminaison. Soit i_1 un indice au delà duquel il n'y a aucune application de $(P_1), (P_2)$. Soit $\mathcal{V}'$ l'ensemble des variables qui sont membre gauche d'une diséquation résolue de $\mathcal{P}_{i_1+1}$. Remarquons alors que, par contrôle imposé à (Ex_p), le multi-ensemble des tailles des positions des variables de $\mathcal{V}'$ a décrû entre deux applications consécutives de (Ex_p).

La preuve de complétude est, comme précédemment, une vérification de la réductibilité des problèmes qui ne sont pas des problèmes sans cycle. Nous la laissons au lecteur. □

4.6 Problèmes de compléments

Les problèmes de complément sont des problèmes équationnels particuliers qui interviennent dans de nombreuses applications (que nous aborderons par la suite). Typiquement, si $t \in T(F,X)$, notons $[\![t]\!]$ l'ensemble des termes fermés qui sont des instances de t. Alors, le calcul du complémentaire de $[\![t']\!]$ dans $[\![t]\!]$ s'exprimera à l'aide d'un problème de complément. On peut en effet dire que $[\![t]\!] - [\![t']\!]$ est l'ensemble des termes $x\sigma$ où σ est une solution dans $T(F)$ du problème:

$$\exists Var(t), \forall Var(t') : x = t \wedge x \neq t'$$

d'inconnue x.

En fait, nous considérerons des problèmes plus généraux où des ensembles infinis de termes fermés sont représentés non seulement par des termes avec variables mais aussi par des *termes contraints* :

Définition 4.50 *Un terme contraint est une paire (t,d) où $t \in T(F,X)$ et d est une conjonction de diséquations de la forme $z \neq u$ où z est une variable de t et u est un sous-terme linéaire de t ne contenant pas z.*

Plus généralement, nous considérerons aussi des ensembles finis de termes contraints ainsi que des problèmes de complément *à des ensembles complets de positions.*

Définition 4.51 *Soit $t \in T(F,X)$. Un* ensemble complet de positions de t *est un sous-ensemble $Q \subseteq Pos(t)$ tel que:*

- $\epsilon \notin Q$
- *Si $x \equiv t/p$ est une variable ayant au moins deux occurrences dans t, alors $p \in Q$*
- $\forall p \in Pos(t) - (Q \cup \{\epsilon\})$, $\exists q \in Q$, $(q \leq p$ *et* $\forall q' \in Q$, $(q' \leq p \Rightarrow q' \leq q)$

Cette définition n'a rien à voir avec les ensembles complets de positions définis dans [Fri86]. Ils serviront essentiellement dans les applications (chapitre 5) à limiter au maximum la taille des problèmes équationnels considérés: il sera suffisant de calculer les compléments à un ensemble complet de positions au lieu de l'ensemble entier.

Exemple 4.8 Si $t = f(t_1,\ldots,t_n)$ est un terme linéaire, alors $Q = \{1,\ldots,n\}$ est un ensemble complet de positions.
Si t est quelconque, l'ensemble de toutes ses positions, sauf la racine, est un ensemble complet de positions.

Si Q_t est un ensemble complet de positions de t, on note $\widehat{Q_t}$ l'ensemble des positions qui précèdent immédiatement:

$$p \in \widehat{Q_t} \quad \text{ssi} \quad p \in Pos(t) \text{ et } \exists i \geq 0, p \cdot i \in Q_t$$

On note $\overline{Q_t}$ la *lisière* de t : $\overline{Q_t}$ est l'ensemble des positions de Q_t qui sont extrémales pour l'ordre lexicographique.

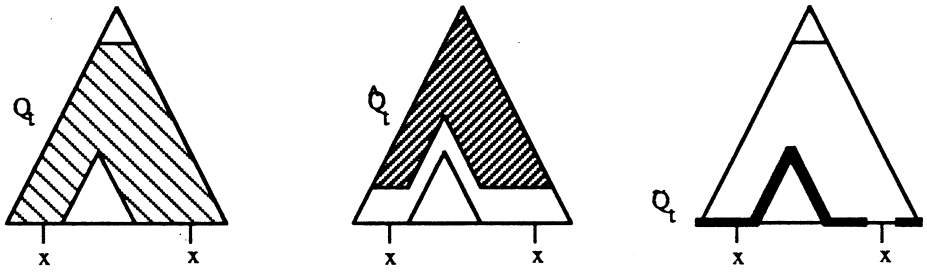

Figure 4.18: Les trois ensembles Q_t, $\widehat{Q_t}$ et $\widetilde{Q_t}$

Enfin, $\widetilde{Q_t}$ est obtenu à partir de $\overline{Q_t}$ en ne répétant pas des positions correspondant à des termes identiques:

$$\{t/p \mid p \in \widetilde{Q_t}\} = \{t/p \mid p \in \overline{Q_t}\} \quad \text{et} \quad \forall p, q \in \widetilde{Q_t},\ t/p \not\equiv t/q$$

$\widetilde{Q_t}$ n'est pas défini de façon unique : pour deux positions p et q de $\overline{Q_t}$ telles que $t/p \equiv t/q$, on peut choisir $p \in \widetilde{Q_t}$ (et $q \notin \widetilde{Q_t}$) ou l'inverse.

Les trois ensembles $Q_t, \widehat{Q_t}$ et $\widetilde{Q_t}$ sont représentés de façon plus explicite sur la figure 4.18.

Définissons maintenant les problèmes de complément:

Définition 4.52 *Soit $\mathcal{L}$ un ensemble fini de termes de $T(F,X)$*[13]*, (t,d) un terme contraint et Q un ensemble complet de positions de t. Le* problème de complément $\mathcal{C}((t,d),\mathcal{L},Q)$ *est le problème équationnel:*

$$\exists \vec{w}, \forall \vec{y}:\ d \wedge \Big(\bigwedge_{q \in \widehat{Q_t}} \bigwedge_{l \in \mathcal{L}} t/q \neq l \Big) \wedge \Big(\bigwedge_{p \in \widetilde{Q_t}} x_p = t/p \Big)$$

où $\vec{w} = Var(t)$, $\vec{y} = Var(\mathcal{L})$[14] *et $\mathcal{I} = \{x_p, p \in Q\}$ est un sous-ensemble de X disjoint de $\mathcal{L} \cup Var(t)$ et tel que, pour tout p, p', $x_p \equiv x_{p'}$ ssi $t/p \equiv t/p'$.*

On note encore $\widehat{t}$ le terme $t[x_1, \ldots, x_m]_{p_1,\ldots,p_m}$ si $\overline{Q_t} = \{p_1, \ldots, p_m\}$. Si bien que, si $\widetilde{Q_t} = \{x_1, \ldots, x_k\}$ (avec $k \leq m$), on peut écrire:

$$t \equiv \widehat{t}\{x_1 \to t/p_1; \ldots; x_k \to t/p_k\}$$

L'objectif de cette section est de donner des propriétés supplémentaires des formes résolues, en tenant compte de la spécificité des problèmes considérés. Ces propriétés sont nécessaires pour obtenir les résultats du chapitre suivant.

Nous utiliserons les règles de la figure 4.19 pour la simplification des problèmes de complément. (Notons $\mathcal{R}_c$ le système ainsi obtenu). Ces règles sont celles du chapitre

[13] $\mathcal{L}$ pour "Left hand side"; dans le chapitre 5 $\mathcal{L}$ sera en effet l'ensemble des membres gauches d'un système de réécriture.

[14] $\mathcal{L}$ et $Var(t)$ sont ici supposés disjoints

3, excepté (Ex_3) qui est la composée de (Ex_1), (R_1) et (MF_2). Elles sont donc toutes adéquates dans $T(F)$. La terminaison de $\mathcal{R}_c$ n'est pas une conséquence du théorème 3.20 parce que nous avons ici privilégié l'explosion par rapport à l'élimination des paramètres (EP_2). Nous utiliserons d'ailleurs explicitement le fait que les problèmes initiaux sont des problèmes de complément dans la preuve de terminaison. Montrons tout d'abord quelques propriétés des problèmes transformés par $\mathcal{R}_c$.

Lemme 4.53 *Si $\mathcal{P}$ est un problème de complément (avec les mêmes notations que dans la définition 4.52) et si $\mathcal{P} \mapsto^*_{\mathcal{R}_c} \mathcal{P}''$, alors $\mathcal{P}''$ est de la forme*

$$\exists \vec{w''}, \forall \vec{y''} : \bigwedge_{p \in \widetilde{Q}} x_p = u''_p \bigwedge_{i=1,\ldots,r''} d''_i$$

où

- $\forall\, j$, d''_j *est une disjonction de diséquations*
- $Var(d''_1, \ldots, d''_{r''}) \cap Inc(\mathcal{P}'') \subseteq Var(u''_1, \ldots, u''_m) = Inc(\mathcal{P}'')$
- $x_1, \ldots, x_m$ *n'ont qu'une occurrence dans* $\mathcal{P}''$
- *pour tout* p, u''_p *est un terme linéaire et* $Var(u''_p) \cap Var(u''_q) = \emptyset$ *si* $p \neq q$
- *pour tout* j *et toute diséquation* $u \neq v$ *de* d''_j, *-ou bien* $Param(u) = \emptyset$ *-ou bien* $Param(v) = \emptyset$

Preuve

Il suffit de remarquer que, si $\mathcal{P}_1$ est de la forme énoncée dans le lemme et si $\mathcal{P}_1 \mapsto \mathcal{P}_2$, alors $\mathcal{P}_2$ est de la forme énoncée dans le lemme. On termine par récurrence sur la longueur de la chaine de transformations; la seule propriété qui n'est pas trivialement satisfaite initialement est $Var(u''_p) \cap var(u''_q) = \emptyset$. Elle découle de la définition d'un ensemble complet de positions : si x est une variable ayant deux occurrences dans t, alors les positions de x appartiennent à Q et une seule d'entre elle est dans $\widetilde{Q}$. □

Lemme 4.54 *$\mathcal{R}_c$ est à terminaison finie lorsqu'appliqué aux problèmes de complément.*

Preuve

Il suffit pour cette preuve d'inverser l'ordre des deux premières fonctions d'interprétation dans la preuve du théorème 3.20:

- $\phi_1(\exists \vec{w}, \forall \vec{y} : d_1 \wedge \ldots \wedge d_n)$ est le multi-ensemble $\{Tpp(d_1), \ldots, Tpp(d_n)\}$ des multi-ensembles des tailles des positions des paramètres dans chaque diséquation (ou équation) de d_i.
- $\phi_2(\exists \vec{w}, \forall \vec{y} : d_1 \wedge \ldots \wedge d_n)$ est le multi-ensemble des nombres de paramètres dans chaque disjonction
- $\phi_3(\mathcal{P})$ est le nombre de diséquations de $\mathcal{P}$ de la forme $w \neq t$ où w est une inconnue et t est un terme non variable contenant une occurrence de paramètre

$$(EP_1) \quad \forall \vec{y}, y : P \quad \mapsto \quad \forall \vec{y} : P$$

Si $y \notin Var(P)$.

$$(EP_2) \quad \forall \vec{y}, y : P \wedge (y \neq t \vee d) \quad \mapsto \quad \forall \vec{y}, y : P \wedge d\{y \rightarrow t\}$$

Si $y \notin Var(t)$ et aucune autre règle autre que (Nc) n'est applicable

$$(Nc) \quad \forall \vec{y} : P \wedge (d_1 \vee d_2) \quad \mapsto \quad \forall \vec{y} : P \wedge d_1$$

Si $\vec{y} \cap Var(d_1) = \emptyset$ ou $\vec{y} \cap Var(d_2) = \emptyset$.

$$(D_2) \quad f(t_1, \ldots, t_n) \neq f(u_1, \ldots, u_n) \quad \mapsto \quad t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n$$

$$(I_2) \quad f(t_1, \ldots, t_n) \neq g(u_1, \ldots, u_m) \quad \mapsto \quad \top$$

Si $f \neq g$.

$$(O_2) \quad z \neq t \quad \mapsto \quad \top$$

Si $z \in Var(t)$ et $z \not\equiv t$.

$$(T_2) \quad t \neq t \quad \mapsto \bot$$

$$(Ex_3) \quad \exists \vec{w}, w : \mathcal{P}[w \neq t] \quad \mapsto \quad \exists \vec{w}, w_1, \ldots, w_m : \mathcal{P}\{w \rightarrow f(w_1, \ldots, w_m)\}$$

Si

1. t contient une occurrence de paramètre et n'est pas lui-même un paramètre
2. $f \in F$ et $w_1, \ldots, w_m \notin Var(\mathcal{P})$
3. aucune autre règle autre que $(N_c), (EP_2)$ n'est applicable

Figure 4.19: Règles de transformation des problèmes de complément

- $\phi_4(\mathcal{P})$ est la taille de $\mathcal{P}$.
- $\phi_5(\mathcal{P})$ est le nombre de paramètres de $\mathcal{P}$.

(EP_1) fait décroitre ϕ_5 et ne modifie pas les autres fonctions d'interprétation. (EP_2) fait décroitre ϕ_2. Cette règle ne fait pas croitre ϕ_1 (et nous utilisons là le fait que les problèmes de départ sont des problèmes de complément). En effet, d'après le lemme 4.53, si $y \neq t$ est une diséquation d'un problème $\mathcal{P}$ obtenu par transformations d'un problème de complément, alors t ne contient pas d'occurrence de paramètre.

La règle (D_2), si elle est apliquée à des termes faisant intervenir des occurrences de paramètres, fait strictement décroitre ϕ_1. Sinon, elle ne modifie pas ϕ_3 et fait décroitre ϕ_4.

Ainsi, toutes les règles font décroitre la composée lexicographique $(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5)$ (la vérification est triviale pour les règles qui n'ont pas été mentionnées ci-dessus). Ce qui prouve la terminaison. □

Il nous reste à prouver une propriété de complétude. Définissons donc d'abord les formes résolues qui nous intéressent ici.

Si $\mathcal{P} = \mathcal{C}((t,d),\mathcal{L},Q)$, on note $IR(\mathcal{P})$ l'ensemble des formes irréductibles de $\mathcal{P}$ pour $\mathcal{R}_c$. On notera aussi parfois $h(t)$ ("hauteur" de t) le nombre profondeur(t).

Définition 4.55 *$\mathcal{P}' \in IR(\mathcal{C}((t,d),\mathcal{L},Q))$ où Q est un ensemble complet de positions de t est dit* simple *si $\mathcal{P}' \equiv \bot$ ou bien*

$$\mathcal{P}' \equiv \exists \vec{w'} : \bigwedge_{p \in \widetilde{Q}} (x_p = u'_p \wedge d_p) \bigwedge_{i=1,\ldots,k} z_i \neq v_i$$

avec les propriétés suivantes:

1. *$x_1, \ldots, x_p$ n'ont qu'une occurrence dans $\mathcal{P}'$*
2. *pour tout p, (u'_p, d_p) est un terme contraint*
3. *pour tout $i = 1, \ldots, k$, z_i est une variable et $z_i \notin Var(v_i)$*
4. *pour tout p, u'_p est un terme linéaire et, si $p \neq q$, $Var(u'_p) \cap Var(u'_q) = \emptyset$*
5. *$Var(z_1, \ldots, z_k, v_1, \ldots, v_k) \subseteq Var(u'_1, \ldots, u'_m)$ $(m = |\widetilde{Q}|)$*
6. *pour tout i, $z_i \in Var(u'_p) \Rightarrow Var(u'_p) \cap Var(v_i) = \emptyset$*
7. *$k \leq \alpha\beta + |d|$ où α est le cardinal de $\widehat{Q}$, β le nombre de termes non linéaires dans $\mathcal{L}$ et $|d|$ est le nombre de diséquations de d.*
8. *pour tout p, $h(u'_p) \leq max(h(t/p), max_{l \in \mathcal{L}} h(l))$*
9. *si t est linéaire, alors, pour tout i, v_i est un terme linéaire*

Notre objectif n'est autre que de prouver la complétude de $\mathcal{R}_c$ vis-à-vis de $\mathcal{F}_I$ ensemble des problèmes de complément et $\mathcal{F}_R$ ensemble des problèmes simples. Pour cela, nous ayons à établir un certain nombre de résultats techniques préliminaires. Dans toute la suite, nous supposerons $(t,d), \mathcal{L}$ et Q (ensemble complet de positions de t) fixés. On note $\mathcal{P}$ le problème $\mathcal{C}((t,d),\mathcal{L},Q)$ et $\mathcal{P}' \in IR(\mathcal{P})$ est un problème différent de $\bot$.

Lemme 4.56 *$\mathcal{P}'$ est de la forme*

$$\exists \vec{w'} : \bigwedge_{p \in \widetilde{Q}} x_p = u'_p \bigwedge_{i=1,\ldots,r} z_i \neq v_i$$

avec

- *$x_1, \ldots, x_p$ n'ont qu'une occurrence dans $\mathcal{P}'$*
- *pour tout i, $z_i \notin Var(v_i)$*
- *$Var(z_1, \ldots, z_r, v_1, \ldots, v_r) \subseteq Var(u'_1, \ldots, u'_m)$*
- *pour tout p, u'_p est linéaire et, si $p \neq q$, alors $Var(u'_p) \cap Var(u'_q) = \emptyset$.*

Preuve
D'après le lemme 4.53, $\mathcal{P}'$ est de la forme

$$\exists \vec{w'} : \bigwedge_{p \in \widetilde{Q}} x_p = u'_p \bigwedge_{i=1,\ldots,r} z_i \neq v_i$$

où $x_1, \ldots, x_m$ n'ont qu'une occurrence dans $\mathcal{P}'$ et $Var(u'_p) \cap Var(u'_q) = \emptyset$ pour $p \neq q$.

$\mathcal{P}'$ ne contient pas de paramètre car, si c'était le cas, l'une des règles de $\mathcal{R}_c$ serait applicable. (La vérification en est laissée au lecteur).

$z_i \notin Var(v_i)$ sinon (T_2) ou (O_2) est applicable.

La condition $Var(z_1, \ldots, z_r, v_1, \ldots, v_r) \subseteq Var(u_1, \ldots, u_m)$ est quant à elle une conséquence du lemme 4.53. □

Lemme 4.57 *Le nombre (r) de diséquations de $\mathcal{P}'$ est inférieur au nombre de diséquations de $\mathcal{P}$.*

Preuve
Si $\mathcal{P}_1 \mapsto_R \mathcal{P}_2[u \neq v]_p$, nous noterons $\phi_{\mathcal{P}_1,\mathcal{P}_2,R}(u \neq v)$[15] la diséquation de $\mathcal{P}_1$ définie comme suit:

- si $\mathcal{P}_1/p \equiv \mathcal{P}_2/p \equiv u \neq v$ alors $\phi_{\mathcal{P}_1,\mathcal{P}_2,R}(u \neq v) \equiv u \neq v$
- si $\mathcal{P}_1[y \neq t \vee u' \neq v' \vee d] \mapsto_{EP_2} \mathcal{P}_2 \equiv \mathcal{P}_1[u \neq v \vee d\{y \to t\}$, alors $\phi_{\mathcal{P}_1,\mathcal{P}_2,R}(u \neq v) \equiv u' \neq v'$
- si $\mathcal{P}_1[f(t_1, \ldots, t_n) \neq f(u_1, \ldots, u_n)] \mapsto_{D_2} \mathcal{P}_2 \equiv \mathcal{P}_1[t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n]$ et que $u \neq v \equiv u_i \neq v_i$, alors $\phi_{\mathcal{P}_1,\mathcal{P}_2,R}(u \neq v) \equiv f(t_1, \ldots, t_n) \neq f(u_1, \ldots, u_n)$
- Si $\mathcal{P}_1 \mapsto_{Ex_3} \mathcal{P}_2$ et $\mathcal{P}_1/p \equiv u' \neq v'$ et $\mathcal{P}_2/p \equiv u \neq v \equiv (u' \neq v')\{w \to f(w_1, \ldots, w_n)\}$, alors $\phi_{\mathcal{P}_1,\mathcal{P}_2,R}(u \neq v) \equiv u' \neq v'$

[15] On sous-entend ici $u \neq v$ "à la position p": une même diséquation qui aurait deux occurrences dans le problème peut alors avoir deux images. Nous aurions dû pour être rigoureux faire figurer la position en paramètre.

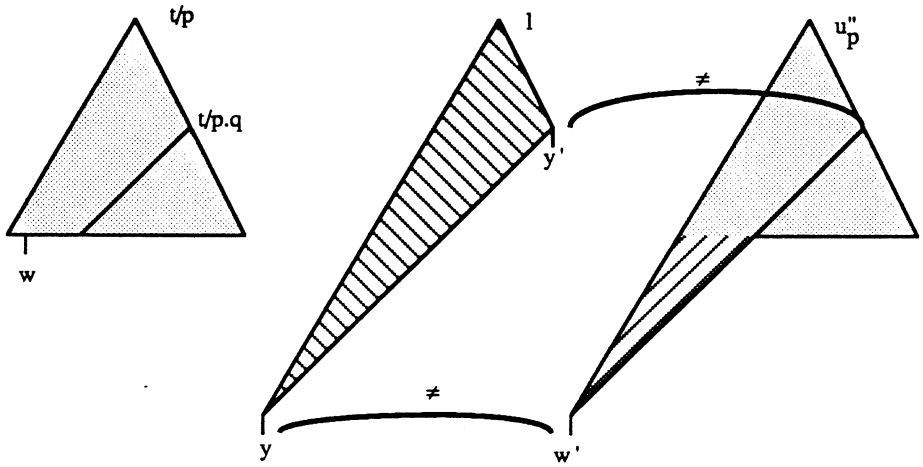

Figure 4.20: Forme générale des diséquations obtenues après explosions et décompositions

$\phi_{\mathcal{P}_1,\mathcal{P}_2,R}$ est injective par construction. En composant ces applications, le long de la chaine de transformation de $\mathcal{P}$ à $\mathcal{P}'$ on obtient alors une application injective de l'ensemble des diséquations de $\mathcal{P}$ dans l'ensemble des diséquations de $\mathcal{P}'$. □

Si $\mathcal{P}_1 \mapsto_{R_1} \mapsto \ldots \mapsto_{R_{n-1}} \mathcal{P}_n$ est une chaine de transformations, le *chemin* issu de $\mathcal{P}_1$ et aboutissant à $u \neq v \equiv \mathcal{P}_n/p$ est la suite définie par récurrence par $u_0 \neq v_0 \equiv u \neq v$ et $u_{i+1} \neq v_{i+1} \equiv \phi_{\mathcal{P}_{n-i-1},\mathcal{P}_{n-i},R_{n-i-1}}(u_i \neq v_i)$.

Lemme 4.58 *Si* $\mathcal{P} \mapsto^*_{\mathcal{R}_c-\{(EP_2)\}} \mathcal{P}''$ *et* $\mathcal{P}''$ *est irréductible pour* $\mathcal{R}_c - \{(EP_2)\}$, *alors*

$$\mathcal{P}'' \equiv \exists \vec{w''}, \forall \vec{y} : \bigwedge_{p \in \widetilde{Q}} x_p = u_p\sigma \bigwedge_{i=1,\ldots,k''} d_i$$

où d_i *est une disjonction de diséquations qui sont de l'une des formes:*

- $z \neq v$ *avec* $Param(z \neq v) = \emptyset$
- $y \neq t\sigma/p \cdot q$ *où* y *est un paramètre et* $p \in \widetilde{Q}$
- $y \neq w\sigma$ *où* y *est un paramètre et* w *est une inconnue auxiliaire*

Preuve

Les différentes situations sont résumées dans la figure 4.20. σ est (informellement) la composée des substitutions $\{w \to f(w_1, \ldots, w_p)\}$ qui sont utilisées dans les règles (Ex_3) le long de la transformation aboutissant à $\mathcal{P}''$.

Lorsqu'une position q d'un paramètre y' de $l \in \mathcal{L}$ est aussi une position de t/p (et qu'aucune incompatibilité n'est survenue), la diséquation $t/p \neq l$ conduira à $t\sigma/p \cdot q \neq y'$. Lorsqu'une position q d'un paramètre y de $l \in \mathcal{L}$ est un suffixe d'une position de t/p, la règle d'explosion est utilisée jusqu'à "faire remonter" le paramètre de position q à la

surface. On trouve alors une diséquation $y \neq w'$. L'équation $x_p = t/p$ a quant à elle été transformée en $x_p = u''_p$ où u''_p est la "réunion" des deux arbres. (cf figure)

Indiquons comment formaliser cela. Tout d'abord, à cause du contrôle, les diséquations de $\mathcal{P}''$ -ou bien ne contiennent pas d'occurrence de paramètre -ou bien sont de la forme $y \neq u$ où y est un paramètre. Alors, u ne contient pas de paramètre d'après le lemme 4.53.

Soit maintenant $t_0 \neq t'_0 \equiv y \neq u, \ldots t_i \neq t'_i, \ldots, t_n \neq t'_n \equiv l \neq t/p$ le chemin aboutissant à $y \neq u$. Pour simplifier, éliminons de ce chemin les termes de la suite qui sont identiques à leurs prédécesseurs. Pour tout $i \leq n-1$, trois situations peuvent se présenter:

1. ou bien $t_i \equiv w$ est une inconnue, $t'_i \equiv l/q_i$, $t_{i+1} \equiv f(w_1, \ldots, w_p)$, $t'_{i+1} \equiv l/q_i$ et $\sigma_{i+1} = \{w \rightarrow f(w_1, \ldots, w_p)\} \circ \sigma_i$ ce qui correspond à un cas d'application de l'explosion.

2. ou bien $t_i \equiv t\sigma_i/p \cdot q_i$, $t'_i \equiv l/q_i$, $t_{i+1} \equiv t\sigma_i/p \cdot q_i \cdot k_i$ et $t'_{i+1} \equiv l/q_i \cdot k_i$. Ce qui correspond à une décomposition.

3. ou bien $t_i \equiv f(w_1, \ldots, w_p)$, $t'_i \equiv l/q_i$, $t_{i+1} \equiv w_j$ et $t'_{i+1} \equiv l/q_i \cdot j$. Ce qui correspond à une décomposition après explosion.

Si l'on se trouve au moins une fois dans le cas 1, la dernière transformation ne peut tomber que dans le troisième cas et $y \equiv l/q$ et $u \equiv w'$ est une variable.

Si l'on ne se trouve jamais dans le cas 1, on reste toujours dans le cas 2 et finalement, $y \equiv l/q$ et $u \equiv t\sigma/p \cdot q$.□

Désormais nous noterons t_0 le terme $\widehat{t}\{x_1 \rightarrow u'_1; \ldots; x_k \rightarrow u'_k\}$ si $\widetilde{Q_t} = \{x_1, \ldots, x_k\}$.

Lemme 4.59 *Toute position de t_0 est comparable à l'un des p_j.*

Preuve
Ce résultat est une conséquence de la définition d'un ensemble complet de positions: toute position de t est comparable à l'un des p_j. □

Lemme 4.60 *Pour tout i, z_i et v_i sont des sous-termes de t_0.*

Preuve
Soit $u \neq v \equiv \phi(z_i \neq v_i)$. Au plus l'un des deux termes u, v contient une occurrence de paramètre. Nous supposerons donc que u ne contient pas de paramètre.

Si $u \neq v$ est une diséquation de d, alors $\phi_{\mathcal{P}_i, \mathcal{P}_{i+1}, R_{i+1}}$ est différente de l'identité seulement si R_{i+1} est (Ex_3) ou (D_2). Dans les deux cas, la propriété que les deux membres de la diséquation soient des sous-termes de t_0 est conservée. On termine dans ce cas par une récurrence sur n, en remarquant que, comme (t, d) est un terme contraint, les membres des diséquations de d sont de sous-termes de t et donc de t_0.

Si $u \neq v$ est une diséquation $t/p \neq l$. Soit alors $\mathcal{P}''$ le problème obtenu comme dans le lemme 4.58. La diséquation $y \neq v''$ de $\mathcal{P}''$ qui est sur le chemin issu de $u \neq v$ et aboutissant à $z_i \neq v_i$ est alors de l'une des formes énoncée dans le lemme 4.58. Dans tous les cas v''

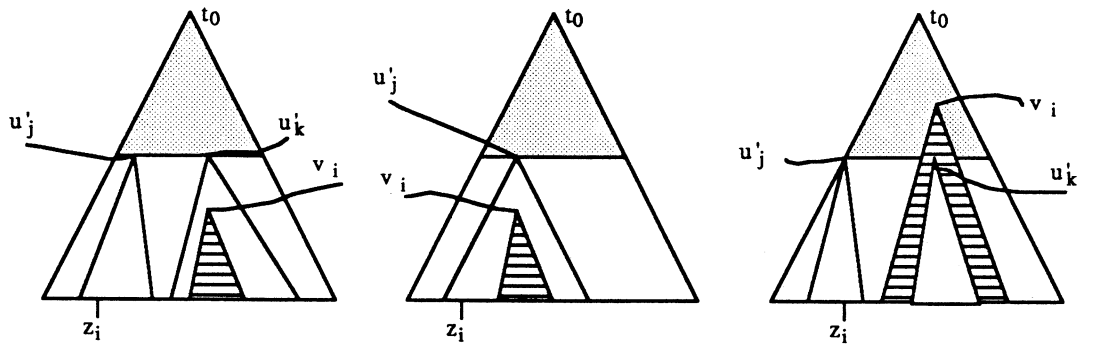

Figure 4.21: Positions relatives de u_j, z_i et v_i

est un sous-terme de t_0 et il nous suffit de faire un raisonnement analogue à celui que nous avons fait dans le cas où $u \neq v$ ne contient pas de paramètre. □

Lemme 4.61 *Si $z_i \in var(u'_p)$, alors -ou bien v_i est un sous-terme de u'_p -ou bien $Var(v_i) \cap Var(u'_p) = \emptyset$.*

Preuve
D'après le lemme 4.60, z_i et v_i sont des sous-termes de t_0 et, d'après le lemme 4.59 ils sont comparables à l'un des u'_j. Comme $z_i \notin Var(v_i)$, trois situations peuvent alors se produire. Elles sont représentées dans la figure 4.21.

On voit aisément sur cette figure que les conclusions du lemme sont satisfaites dans chacun des trois cas puisque, d'après le lemme 4.56, u'_j et u'_k n'ont de variable commune que si $u'_j \equiv u'_k$. □

Pour chaque p, soit d_p la conjonction des diséquations $z_i \neq v_i$ de $\mathcal{P}'$ telles que $Var(z_i, v_i) \subseteq Var(u'_p)$. On déduit alors du lemme précédent que $\mathcal{P}'$ s'écrit

$$\mathcal{P}' \equiv \exists \vec{w'} : \bigwedge_{p \in \tilde{Q}} (x_p = u'_p \wedge d_p) \bigwedge_{i=1,\ldots,r} z_i \neq v_i$$

où $z_i \in var(u'_p) \Rightarrow Var(v_i) \cap Var(u'_p) = \emptyset$.

Lemme 4.62 *Pour tout $p \in \tilde{Q}$, (u'_p, d_p) est un terme contraint.*

Preuve
u'_p étant linéaire d'après le lemme 4.56, il suffit de vérifier que, si $z \neq u$ est une diséquation de d_p, alors z et u sont des sous-termes de u'_p. Sachant que $Var(z, u) \subseteq Var(u'_p)$, ce résultat se voit sur la figure 4.21.□

Lemme 4.63 *$r \leq \alpha\beta + |d|$ où α est le cardinal de $\widehat{Q}$, β est le nombre de termes non linéaires dans $\mathcal{L}$ et $|d|$ est le nombre de diséquations de d.*

Preuve
Remarquons tout d'abord que $r \leq \alpha|\mathcal{L}| + |d|$ d'après le lemme 4.57. Il suffit alors de montrer que toute diséquation entre un terme et un terme linéaire universel est éliminée. Si l'on considère la disjonction d_i du lemme 4.58 qui contient les extrémités des chemins issus de $l \neq t/p$ où l est un terme linéaire de $\mathcal{L}$, chaque paramètre y apparait au plus une fois. L'élimination des paramètres ne permet pas ainsi de faire apparaitre de diséquation entre termes sans paramètres: toute diséquation issue de $l \neq t/p$ contient au moins un paramètre. Comme tous les paramètres sont éliminés dans $\mathcal{P}'$, il en résulte qu'il n'y a pas de chemin de $l \neq t/p$ à une diséquation de $\mathcal{P}'$. □

Lemme 4.64 *Pour tout p, $h(u'_p) \leq max(h(t/p), max_{l \in \mathcal{L}} h(l))$.*

Preuve
La figure 4.20 illustre bien ce résultat: si $x_p = u''_p$ est une équation d'un problème obtenu par transformations de $\mathcal{P}$, les positions de u''_p sont contenues dans $Pos(t/p) \bigcup_{l \in \mathcal{L}} Pos(l)$. D'où le résultat. □

Lemme 4.65 *Si t est linéaire, v_i est linéaire pour tout i.*

Preuve
Ceci est une conséquence du fait que les transformations autres que le remplacement, ne font intervenir que les termes d'une même disjonction. Si une inconnue z n'apparait qu'une fois (au plus) dans chaque terme de la disjonction, cette propriété reste ainsi valide tout au long de la transformation. □

On obtient enfin le théorème de complétude:

Théoreme 4.66 *Toute forme irréductible (pour $\mathcal{R}_c$) d'un problème de complément est un problème simple.*

Preuve
Les propriétés 1,3,4,5 des problèmes simples sont une conséquence du lemme 4.56. La propriété 2 résulte du lemme 4.62. Les propriétés 6,7,8,9 sont fournies repectivement par les lemmes 4.61, 4.63, 4.64,4.65. □

Chapitre 5

Application à la la réductibilité inductive

Introduction

Dans ce chapitre, nous allons utiliser les résultats de chapitres précédents pour la décision de la complétude suffisante des spécifications algébriques [EKP78,GHM78] ainsi que pour la réductibilité inductive [JK86b]. La méthode développée ici possède aussi d'autres applications exposées dans le prochain chapitre et dans [Com88].

Définition 5.1 *Un symbole $f \in F$ est* bien défini *par rapport à $C \subseteq F$ dans une spécification (multi-sorte) (S, F, E) si:*

$$\forall t_1, \ldots, t_n \in T(F), \exists t \in T(C), f(t_1, \ldots, t_n) =_E t$$

Ce genre de propriété est utilisé dans au moins deux domaines:

La protection des spécifications algébriques

Quand on utilise des définitions hiérarchiques de types [Ber79,EKP78], on souhaite qu'une nouvelle définition vienne s'ajouter aux précédentes sans les pertuber. Plus précisément, soient $\Sigma = (S, F, E)$ une spécification et $\Sigma_2 = (S \cup S', F \cup F', E \cup E')$ une spécification contenant la première. On dit alors que:

- Σ_2 est *cohérente par rapport à* Σ_1 si :

$$\forall t, t' \in T(F), (t =_{E \cup E'} t') \Leftrightarrow (t =_E t')$$

- Σ_2 est *suffisamment complète* par rapport à Σ_1 si:

$$\forall t \in T(F \cup F'), (sort(t) = \underline{s} \in S) \Rightarrow (\exists t' \in T(F), t =_{E \cup E'} t')$$

- Σ_2 *protège* Σ_1 si elle est cohérente est suffisament complète par rapport à Σ_1.

Ces définitions sont données dans [Ber79,EKP78,GTW78,GHM78,MG85] à quelques variantes près. Intuitivement, une spécification est cohérente si l'on n'identifie pas deux termes qui étaient distincts dans les types déjà définis. Une spécification est suffisamment complète si l'on ajoute pas de "nouveau" terme dans les types déjà définis. Ces propriétés s'expriment formellement en termes de morphismes dans la théorie des types abstraits algébriques [MG85].

La propriété de complétude suffisante est équivalente, avec nos définitions, à la bonne définition des symboles de F' dont le codomaine est une sorte de S.

Exemple 5.1 $F = \{\ 0 : \rightarrow \underline{int};\ \ p, s\ \underline{int} \rightarrow \underline{int};\ \ eq :\ \underline{int} \times \underline{int} \rightarrow \underline{bool}\ \}$

$$E = \{\ \begin{array}{lll} s(p(x)) == x; & p(s(x)) == x; & eq(x,x) == true; \\ eq(x,s(x)) == false; & eq(s(x),x) == false; & eq(s(x),s(y)) == eq(x,y); \\ eq(x,p(x)) == false; & eq(p(x),x) == false; & eq(p(x),p(y)) == eq(x,y) \end{array}\ \}$$

Nous notons ici $==$ l'égalité des équations utilisées dans les axiomes.

On s'intéresse au problème suivant : cette spécifiaction protége-t-elle la spécification des booléens ? (Nous supposons les booléens déjà spécifiés. Leur définition importe peu ici. Notons seulement qu'il y a deux classes d'équivalence distinctes (et deux seulement): celle de $true$ et celle de $false$).

Ici, la spécification est cohérente (par rapport aux booléens) ssi on ne peut déduire de E l'égalité $true = false$. La complétude suffisante (par rapport aux booléens) est équivalente à la bonne définition de "eq".

Remarquons qu'en orientant les équations de gauche à droite, nous obtenons un système de réécriture canonique (la terminaison est évidente et la confluence résulte de l'absence de paire critique). Comme de plus $true$ et $false$ sont irréductibles pour ce système de réécriture, $true = false$ n'est pas une équation valide dans la théorie équationnelle définie par E. Par conséquent cette spécification est cohérente (par rapport aux booléens).

Par contre, eq n'est pas bien défini parce que $eq(0, s(s(0)))$ est un terme de sorte $\underline{bool}$, est irréductible et n'est ni $true$, nis $false$. Nous avons donc introduit avec cette spécification un "troisième booléen". Si bien que la spécification n'est pas suffisamment complète (par rapport aux booléens).

Ce résultat n'est, en fait, pas étonnant puisqu'il est impossible d'obtenir une spécification cohérente des entiers relatifs qui protège les booléens sans introduire de nouveau symbole de fonction (cf chapitre 6).

Preuves par induction

Si $t, u \in T(F, X)$, nous dirons que $t = u$ est un théorème inductif lorsque

$$T(F)/=_E \models t = u$$

Le raisonnement équationnel n'est plus complet pour les preuves par induction. Donnons un exemple simple:

Exemple 5.2 S est réduit à une sorte, $F = \{0, s\}$ et $E = \{s(s(0)) == 0\}$. $s(s(x)) = x$ est un théorème inductif en effet, si $t \equiv 0$, alors $s(s(t)) =_E t$, et, si $s(s(t)) =_E t$, alors $s(s(s(t))) =_E s(t)$. Cette preuve est une preuve *par induction structurelle.* On obtient aussi le même résultat en notant que tous les termes de $T(F)$ sont de la forme $s^k(0)$ et en raisonnant par récurrence sur k. Dans tous les cas, nous utilisons explicitement qu'il n'y a pas d'autres termes dans $T(F)$ que ceux que l'on peut construire avec les symboles de F.

Par contre $s(s(x)) = x$ n'est pas un théorème équationnel. En effet, $s(s(0)) \rightarrow 0$ est un système de réécriture canonique. Deux termes sont alors égaux modulo E ssi leurs formes irréductibles sont syntaxiquement égales. Or $s(s(x))$ et x sont irréductibles et distincts.

Pour effectuer (automatiquement) des preuves par induction, certains auteurs ont proposé une méthode s'apparentant à la résolution en démonstration automatique : la méthode de preuve par défaut de cohérence [Mus80,Gog80,HH82,JK86b,Bac88]. Il est bien connu ([Gal86] par exemple) qu'un ensemble de formules du premier ordre A duquel on ne peut déduire une contradiction $(\phi \wedge \neg\phi)$ possède au moins un modèle. Lorsque les formules de A sont des formules fermées (i.e. sans variable libre) en forme prénexe ne contenant pas de quantificateur existentiel on peut même affirmer que, s'il existe un modèle de A, alors il existe un modèle canonique (ou de Herbrand) de A, c'est-à-dire un modèle de la forme $T(F)/=_A$ où $=_A$ est une congruence[1].

Dans la suite, nous appelerons *formule pure* toute formule équationnelle fermée en forme prénexe ne contenant pas de quantificateur existentiel. Un *modèle canonique* d'un ensemble de formules pures A est alors un quotient de $T(F)$ par une congruence $=_A$ qui est un modèle de A.

Le principe des preuves par défaut de cohérence est alors simple : On se donne un ensemble de formules pures A qui possède pour seul modèle canonique (à isomorphisme près) $T(F)/=_E$[2]. Si $u = v$ est un théorème à prouver, on cherche à déduire une contradiction de $A \cup \{u = v\}$. Si l'on déduit une contradiction, $u = v$ n'est pas un théorème inductif. Si au contraire on peut prouver que $A \cup \{u = v\}$ est non contradictoire, alors $u = v$ est un théorème inductif. En effet, $A \cup \{u = v\}$ possède alors un modèle canonique comme nous l'avons rappelé ci-dessus, et ce modèle ne peut être que l'algèbre initiale $T(F)/=_E$.

L'ensemble A d'axiomes est constitué des axiomes équationnels de E et d'un ensemble d'axiomes permettant de restreindre la classe des modèles. Ceux-ci ne sont généralement pas présentés sous forme axiomatique, mais comme un ensemble de formules desquelles on peut déduire une incohérence. Les articles cités plus haut diffèrent ainsi dans les hypothèses effectuées sur la spécification (S, F, E) et dans la manière de déduire une contradiction.

Par exemple, dans [Mus80], on suppose qu'un prédicat d'égalité est complètement défini dans la spécification[3]. Une contradiction est alors produite lorsque l'algorithme de complétion engendre $true = false$. Ce qui revient à choisir $A = E \cup \{true \neq false\}$.

---

[1] Nous ne précisons pas ici l'interprétation des symboles de prédicat autres que l'égalité.

[2] En général, il subsiste des modèles "non standard", d'après le théorème d'incomplétude de Gödel.

[3] Plus formellement, on suppose qu'il existe un symbole de fonction $eq : \underline{s} \times \underline{s} \rightarrow \underline{bool}$ tel que $\forall t, u \in T(F)$, $t =_E u \Leftrightarrow eq(t, u) =_E true$ et $t \neq_E u \Leftrightarrow eq(t, u) =_E false$.

Dans [HH82], on suppose que F est scindé en deux ensembles disjoints C (constructeurs) et D (symboles définis). tels que tout élément de D est bien défini par rapport à C et il n'existe pas d'identité $u =_E v$ entre termes de $T(C)$. Si bien que $T(C)$ est isomorphe à $T(F)/=_E$. Alors, une contradiction est dérivée lorque l'agorithme de complétion engendre une équation $u = v$ entre deux termes de $T(C, X)$. Cela revient à choisir comme ensemble A

$$E \cup \{\forall t_1, \ldots, t_n, u_1, \ldots, u_m, f(t_1, \ldots, t_n) \neq g(u_1, \ldots, u_m) \mid f, g \in C, f \neq g\}$$
$$\cup \{\forall t_1, \ldots, t_n, u_1, \ldots, u_n, (f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n) \Leftrightarrow t_1 = u_1 \wedge \ldots \wedge t_n = u_n) \mid f \in C\}$$

Dans [JK86b], les équations de E sont supposées être orientées en un système de réécriture confluent sur les termes de $T(F)$[4]. Une contradiction est alors produite lorsque l'algorithme de complétion engendre une équation $u = v$ où $u > v$ (pour un ordre de simplification $>$ qui contient la relation de réduction associée au système de réécriture) et u n'est pas *inductivement réductible*:

Définition 5.2 *Soit $\mathcal{R}$ un système de réécriture sur $T(F, X)$ et $t \in T(F, X)$. t est dit* inductivement réductible *ssi, pour tout $T(F)$-substitution σ de domaine $Var(t)$, $t\sigma$ est réductible par $\mathcal{R}$.*

A nouveau, il est possible de donner un ensemble A de formules pures correspondant à cette approche. Celui-ci ne rendrait pas l'exposé plus clair, c'est pourquoi nous ne le reproduisons pas ici. Notons seulement que tous les articles sur les preuves par défaut de consistence pourraient se résumer à donner un ensemble A de formules pures et prouver qu'il n'a qu'un modèle canonique à isomorphisme près.

Convertibilité

Nous avons évoqué le rôle central de la propriété de bonne définition. Malheureusement elle est indécidable [GH78]. On essaye alors de procéder comme pour la décision de l'égalité. L'idée consiste en effet à remplacer la congruence $=_E$ par la relation de réduction $\rightarrow_{\mathcal{R}}$ obtenue par orientation des équations de la théorie. La bonne définition est alors remplacée par la *convertibilité.*

Définition 5.3 *Soit (S, F, E) une spécification, $C \subseteq F$ et $\mathcal{R}$ un système de réécriture tel que $=_E \equiv \leftrightarrow^*_{\mathcal{R}}$. $f \in F$ est dit* convertible à C *ssi :*

$$\forall t_1, \ldots, t_n \in T(F), \exists t \in T(C), f(t_1, \ldots, t_n) \rightarrow^*_{\mathcal{R}} t$$

Dans les théories équationnelles, la décision de l'égalité peut être obtenue facilement lorsque le système de réécriture associé est convergent (cf chapitre 2). Hélas, cette hypothèse de convergence se révèle insuffisante pour assurer l'équivalence entre convertibilité et bonne définition. En effet, comme montré dans [Pla85,KNZ86], la convertibilité est décidable lorsque le système de réécriture est convergent. Au contraire, la bonne définition est indécidable, même dans ce cas [KNZ86]. Néanmoins, il est facile de donner des hypothèses qui permettent d'obtenir l'équivalence (comme nous le verrons plus loin). Le

[4]Cette hypothèse est aussi effectuée dans les autres mehodes. Elle est nécessaire pour pouvoir utiliser l'algorithme de complétion comme base de la déduction équationnelle.

problème de la complétude suffisante (et celui de la bonne définition) se trouvent alors ramenés à un problème de convertibilité.

Remarquons que la convertibilité n'est, en fait qu'un cas particulier de réductibilité inductive: si $\mathcal{R}$ est un système de réécriture canonique et si $F = D \cup C$, tout élément de D est convertible à C ssi $\forall f \in D$, $f(x_1, \ldots, x_n)$ est inductivement réductible.

Décision de la réductibilité inductive

La réductibilité inductive a été prouvée décidable dans le cas général par D. Plaisted [Pla85]. D. Plaisted prouve qu'un terme t est inductivement réductible ssi tous les termes d'un ensemble *fini* d'instances closes de t sont réductibles. (Cet ensemble est appelé *ensemble test*). Malheureusement, cet ensemble test est toujours gigantesque. Par exemple, lorsque $\mathcal{R}$ est réduit à une règle simple : $\mathcal{R} = \{s(s(0)) \rightarrow 0\}$, l'ensemble test correspondant à la décision de la réductibilité inductive de $s(s(x))$ a pour cardinal[5] $5 * 2^{28}$.

L'objet de ce chapitre est de montrer une autre approche permettant de décider de la réductibilité inductive: t est inductivement réductible ssi l'intersection entre le langage des termes fermés qui sont des instances de t et le langage des termes fermés irréductibles est vide. Nous allons donc montrer comment construire une grammaire (ou un automate) engendrant l'ensemble des termes fermés irréductibles qui sont des instances de t. Le nettoyage d'une telle grammaire permet de savoir si le langage engendré est vide et donc si t est inductivement réductible.

La méthode de Plaisted fait appel à des résultats s'apparentant au "pompage" dans les langages hors contexte, ce qui suggérait une approche "langage formel". D'autre part, il est bien connu que les méthodes de pompage, bien que permettant la décision du vide dans certains cas, sont beaucoup moins efficaces que les méthodes par nettoyage d'une grammaire. C'est pourquoi, bien que ne donnant aucun résultat de complexité, nous pouvons espérer que notre approche -d'une part clarifie le problème en comprenant son essence -d'autre part permette d'aboutir à des algorithmes plus efficaces en pratique.

Les grammaires ainsi calculées permettent aussi d'obtenir d'autres résultats, par exemple sur les transformations de spécifications [Com88a] ou sur la compilation du filtrage [Sch88a].

En quoi les problèmes équationnels interviennent dans ces questions ? C'est ce que nous allons montrer dans la section qui suit.

5.1 Réductibilité inductive et problèmes équationnels

Dans tout ce chapitre, nous supposerons donnés un ensemble de symboles de fonction F et un ensemble fini E d'équations $g = d$. Nous noterons $\mathcal{R}$ le système de réécriture obtenu en orientant ces équations : $\mathcal{R} = \{g \rightarrow d \mid g = d \in E\}$. $" \rightarrow "$ désignera aussi la relation de réduction associée à $\mathcal{R}$ et $LHS = \{g_1, \ldots, g_k\} = \{g \mid g \rightarrow d \in \mathcal{R}\}$. On

[5] D'après [KNZ85], article dans lequel l'algorithme de Plaisted est pour la première fois effectivement présenté comme une technique de décision de la réductibilité inductive.

suppose que pour deux indices i et j distincts, $Var(g_i)$ et $Var(g_j)$ sont disjoints[6] et l'on note $Var(LHS) = \bigcup_{i=1,\ldots,k} Var(g_i)$. Enfin, NF est l'ensemble des termes de $T(F)$ qui ne peuvent être réduits par $\rightarrow$.

L'idée (déjà présente dans [Thi84]) est la suivante: un terme t n'est pas réductible à la racine s'il n'est pas filtré par un membre gauche de règle de réécriture. Lorsqu'on s'intéresse à la convertibilité, le "à la racine" ci-dessus peut même être évité. Si $Var(LHS) = \{y_1, \ldots, y_m\}$, f est convertible si et seulement si le problème

$$\forall y_1, \ldots, y_m : g_1 \neq f(x_1, \ldots, x_n) \wedge \ldots \wedge g_k \neq f(x_1, \ldots, x_n)$$

n'a pas de solution dans NF. En effet, il existe une solution dans NF si et seulement s'il existe un terme $t \equiv f(t_1, \ldots, t_n)$ où $t_1, \ldots, t_n$ sont des termes fermés irréductibles qui n'est pas réductible à la racine. Ce qui est équivalent à la non convertibilité de f.

C'est une méthode basée sur ces idées que nous allons développer ici.

5.1.1 Bonne définition et convertibilité

Nous venons de donner une idée du lien existant entre les problèmes de complément et la convertibilité. Voyons ici sous quelle conditions la propriété de bonne définition se ramène à la convertibilité.

Définition 5.4 *Soit $F = C \cup D$. D est dit* stable *pour $\mathcal{R}$ si, pour toute règle $f(t_1, \ldots, t_n) \rightarrow g(u_1, \ldots, u_m)$ de $\mathcal{R}$,*

$$g \in D \Rightarrow f \in D$$

Cette propriété permet d'assurer l'équivalence des notions de bonne définition et de convertibilité[7]:

Proposition 5.5 *Supposons que $F = C \cup D$ et que D est stable par $\mathcal{R}$. Supposons de plus $\mathcal{R}$ convergent. Alors tout opérateur de D est bien défini par rapport à C si et seulement si tout opérateur de D est convertible à C.*

Preuve

$\Rightarrow$ On suppose que tout opérateur de D est bien défini. Soit $NF(D)$ l'ensemble des termes de $T(F)$ dont la racine est dans D et dont la forme normale n'est pas dans $T(C)$. Il nous faut montrer que $NF(D)$ est vide.

Raisonnons par l'absurde et supposons que $t \in NF(D)$. La forme normale t' de t contient donc une occurrence de $f \in D$. Soit alors t'' un sous-terme de t' dont la racine est f. t'' est lui-aussi irréductible.

Si maintenant u vérifie $t'' =_E u$, cela entraine que t'' et u ont même forme normale et donc que $u \rightarrow^* t''$ (puisque t'' est irréductible). D'autre part, D étant stable, si

[6] Cette supposition n'est évidemment pas restrictive puisque les équations de E sont implicitement universellement quantifiées.

[7] Une condition plus générale (mais difficile à tester car non syntaxique) est donnée dans [JK86b]: "la forme normale d'un terme de $T(C)$ est dans $T(C)$".

$v \in T(C)$ et $v \rightarrow^* w$, alors w n'a pas pour racine un élément de D. Il en résulte que, si $t'' =_E u$, alors $u \notin T(C)$.

Cela prouve que f n'est pas bien défini puisque t'' a pour racine f et n'est égal modulo E à aucun élément de $T(C)$. Ce qui est absurde.

$\Leftarrow$ réciproquement, tout symbole de fonction f qui est convertible à C est bien défini puisque

$$(u \rightarrow^* v) \Rightarrow (u =_E v)$$

□

5.1.2 Réductibilité inductive et problèmes de compléments

Nous utilisons ici et dans la suite les définitions et notations introduites dans la section 4.6.

Proposition 5.6 *Un terme $t \in T(F,X)$ est inductivement réductible ssi le problème de complément $C(t, LHS, Q_t)$ n'a pas de solution dans NF. (Q_t étant ici un ensemble complet de positions de t).*

Preuve

Notons $\overline{Q_t} = \{p_1, \ldots, p_m\}$. Soit $\widetilde{Q_t} = \{p_1, \ldots, p_k\}$. Pour chaque $i > k$, il existe un unique indice $j \leq k$ tel que $t/p_j \equiv t/p_i$. $x_1, \ldots, x_k$ sont les inconnues du problème de complément. t est une instance de $\hat{t}$: $t \equiv \hat{t}\{x_1 \rightarrow t/p_1, \ldots, x_k \rightarrow t/p_k\}$. Rappelons que, dans ces conditions, le problème de complément s'écrit:

$$\exists Var(t), \forall \vec{y} : \bigwedge_{p \in \widetilde{Q_t}} x_p = t/p \bigwedge_{q \in \widehat{Q_t}} \bigwedge_{l \in LHS} t/q \neq l$$

où $\vec{y} = Var(LHS)$ et $\widehat{Q_t}$ est l'ensemble des positions non variable de $\hat{t}$.

t est inductivement réductible ssi pour toute $T(F)$-substitution σ de domaine $Var(t)$, $t\sigma$ est réductible à une certaine position p. Ou bien p n'est pas une position de $\widehat{Q_t}$, ce qui signifie que σ n'est pas une substitution à valeurs dans NF. Ou bien $p \in Pos(\widehat{Q_t})$. Dans ce dernier cas, il existe une $T(F)$-substitution θ et un terme $l \in LHS$ tels que $l\theta \equiv t\sigma/p \equiv (t/p)\sigma$. Cela signifie que σ n'est pas solution de la diséquation $l \neq t/p$ (dans laquelle les variables de l sont quantifiées universellement) et donc que $\{x_1 \rightarrow t/p_1; \ldots; x_k \rightarrow t/p_k\}\sigma$ n'est pas solution du problème de complément.

Par conséquent, t est inductivement réductible ssi pour toute $T(F)$-substitution σ de domaine $Var(t)$, $\{x_1 \rightarrow t/p_1; \ldots; x_k \rightarrow t/p_k\}\sigma$ n'est pas solution du problème de complément. Il suffit de remarquer pour terminer que toutes les solutions du problème de complément sont de cette forme. □

On voit donc les liens qui lient inductive réductibilité, bonne définition et problèmes de complément. La difficulté réside maintenant dans le fait que nous n'avons donné dans la section 4.6 qu'un algorithme de résolution des problèmes de complément *dans* $T(F)$. Or d'après la proposition ci-dessus, il nous faut résoudre ces problèmes *dans* NF.

Une méthode envisageable (et nous la développerons d'ailleurs dans la suite) est de procéder "par approximations": au lieu de chercher les solutions dans NF, on cherche en fait les solutions du problème de complément dans $T(F)$, puis à nouveau à l'aide de problèmes de compléments, on recherche parmi ces solutions celles qui sont irréductibles. Quoique, en général, une telle méthode ne termine pas[8], il existe certains cas particuliers pour lesquels on obtient les solutions dans NF de cette manière. Donnons en des exemples.

Exemple 5.3

$$\begin{array}{lcll} F = \{ & 0 :\rightarrow \underline{s}; & p, g, h : \underline{s} \rightarrow \underline{s} & \} \\ \mathcal{R} = \{ & g(0) \rightarrow 0 & h(0) \rightarrow 0 & \\ & h(h(x)) \rightarrow h(x) & g(g(x)) \rightarrow g(x) & \\ & h(p(x)) \rightarrow h(x) & g(h(x)) \rightarrow h(x) & \} \end{array}$$

$\mathcal{R}$ est canonique. La convertibilité de h à $F - \{h\}$ est équivalente à l'absence de solution dans NF au problème:

$$\exists w, \forall y : x = w \wedge h(w) \neq h(0) \wedge h(w) \neq h(h(y)) \wedge h(w) \neq h(p(y))$$

(Si l'on choisit pour ensemble complet de positions de $h(x)$ l'ensemble $Q = \{1\}$ qui est d'ailleurs le seul ensemble complet de positions possible).

Ce problème a pour unique forme résolue $\exists w' : x = g(w')$. La convertibilité de h est donc ramenée à celle de g. Nous sommes alors amenés à résoudre dans NF le problème de complément:

$$\exists w, \forall y : x = w \wedge g(w) \neq g(0) \wedge g(w) \neq g(g(y)) \wedge g(w) \neq g(h(y))$$

Nous n'avons d'ailleurs pas à résoudre ce problème dans NF mais seulement dans l'ensemble des formes irréductibles de $T(F - \{h\})$. Car, s'il n'y a aucune solution irréductible dans $T(F-\{h\})$, toute solution dans NF contient une occurrence de h. Mais nous avons vu que toute instance irréductible de $h(x)$ contient une occurrence de g. Il y a donc contradiction.

Dans le cas présent, la seule forme irréductible du problème est $\exists w' : x = p(w')$, ce qui ramène la convertibilité de g (et donc celle de h) à celle de p. Mais p n'est pas convertible car $p(0)$ est irréductible (ce qui correspond bien sûr à une solution du problème de complément associé). Nous en déduisons que h n'est pas convertible à $F - \{h\}$. Si l'on ajoute par contre la règle $p(0) \rightarrow 0$, $p(x), g(x), h(x)$ sont inductivement réductibles. On obtient ce résultat parce qu'on peut établir une "hiérarchie" sur l'ensemble des symboles fonctionnels. Cette approche est complètement développée dans [Com86]. Nous ne la reprendrons pas complètement ici puisque nous traiterons le cas général.

Mentionnons quand même des cas très simple où la résolution d'un problème de complément permet à elle-seule de résoudre le problème de convertibilité.

5.1.3 Décision de la convertibilté dans des cas "simples"

Nous montrons ici que, lorsqu'il n'y a pas de relation entre les "constructeurs", la résolution d'un problème de complément permet à elle-seule de décider de la convertibilité. Dans ce

[8] Nous en montrerons un raffinement par la suite qui, lui, termine dans le cas général.

cas particulier, des algorithmes de décision sont connus depuis longtemps (voir [Kou85] par exemple).

Nous supposerons donc que F est la réunion de deux ensembles disjoints C et D tels que $T(C) \subseteq NF$ (ce qui est la traduction formelle de "pas de relation entre constructeurs"). Nous supposerons de plus que le système de réécriture est canonique et que D est stable par $\mathcal{R}$. ces hypothèses bien que très fortes sont habituellement effectuées dans des langages comme LPG [BE86] ou FP2 [Jor86].

Proposition 5.7 *Sous les hypothèses mentionnées ci-dessus, tout opérateur de D est bien défini par rapport à C si et seulement si, pour tout $f \in D$, le problème de complément $\mathcal{C}(f(x_1,\ldots,x_n), LHS, \{1,\ldots,n\})$ n'a pas de solution dans $T(C)$.*

Preuve

D'après la proposition 5.5, il suffit de prouver que tout symbole de D est convertible à C. Comme $T(C) \subseteq NF$, d'après la proposition 5.6, il suffit de prouver que, si

$$\mathcal{C}(f(x_1,\ldots,x_n), LHS, \{1,\ldots,n\})$$

n'a pas de solution dans $T(C)$, alors $T(C) = NF$.

Supposons que $T(C) \neq NF$. Alors il existe un terme t dont la racine est $f \in D$ et qui est en forme irréductible. On peut supposer sans perdre de généralité que t ne contient pas d'autre occurrence de symbole de D (il suffit, si ce n'est pas le cas, de remplacer t par un sous terme de t de position maximale et possédant pour racine $f \in D$). t s'écrit alors $f(x_1,\ldots,x_n)\sigma$ où σ est une solution dans $T(C)$ de $\mathcal{C}(f(x_1,\ldots,x_n), LHS, \{1,\ldots,n\})$. Ce qui est contraire à l'hypothèse. □

Remarque

Si l'on s'intéresse à la convertibilité dans ce cas particulier, il n'est pas nécessaire d'utiliser l'algorithme de simplification dans toute sa généralité. D'une part, on ne résout que dans $T(C)$, d'autre part on ne s'intéresse qu'à l'existence d'une solution. Les optimisations que l'on peut ainsi obtenir sont laissées au lecteur intéressé.

Exemple 5.4

$$\begin{array}{lccc} F = \{ & 0 :\rightarrow nat; & succ : nat \rightarrow nat; & +, *, \hat{} : nat \times nat \rightarrow nat \quad \} \\ \mathcal{R} = \{ & 0 + x \rightarrow x & succ(x) + y \rightarrow succ(x + y) & 0 * x \rightarrow 0 \\ & succ(x) * z \rightarrow z + (x * z) & succ(x)\hat{}0 \rightarrow succ(0) & x\hat{}succ(y) \rightarrow x * (x\hat{}y) \quad \} \end{array}$$

$\mathcal{R}$ est canonique et $D = \{+, *, \hat{}\}$ est stable par $\mathcal{R}$. La bonne définition des symboles de D par rapport à $F - D$ est donc équivalente à la réductibilité inductive de $x_1 + x_2$, $x_1 * x_2$, $x_1\hat{}x_2$. Les problèmes

$$\exists w_1, w_2 \forall y_1, y_2 : x_1 = w_1 \wedge x_2 = w_2 \wedge w_1 + w_2 \neq 0 + y_1 \wedge w_1 + w_2 \neq succ(y_1) + y_2$$
$$\exists w_1, w_2 \forall y_1, y_2 : x_1 = w_1 \wedge x_2 = w_2 \wedge w_1 * w_2 \neq 0 * y_1 \wedge w_1 * w_2 \neq succ(y_1) * y_2$$

n'ont pas de solution dans $T(C)$. Comme les quatre premières règles ne font pas intervenir $\hat{}$, on peut en déduire que $+$ et $*$ sont converticles à $C = \{0, succ\}$.

Par contre,

$$\exists w_1, w_2 \forall y_1, y_2 : x_1 = w_1 \wedge x_2 = w_2 \wedge w_1 \hat{} w_2 \neq succ(y_1) \hat{} 0 \wedge w_1 \hat{} w_2 \neq y_1 \hat{} succ(y_2)$$

possède pour (seule) solution dans $T(C)$ $\{x_1 \rightarrow 0; x_2 \rightarrow 0\}$. (Ce que l'on obteint à partir de la seule forme résolue $x_1 = 0 \wedge x_2 = 0$ du problème). Il en résulte que l'exponentiation n'est pas bien définie car iol manque la définition de 0^0.

5.2 Langage des formes normales fermées

Dans la section précédente, nous avons traité le cas où $NF = T(C)$ cas dit "sans relation entre constructeurs". Nous allons montrer ici que, dans le cas général, il est possible de calculer une grammaire engendrant le langage NF. Ce point de vue nous permettra dans les sections suivantes de donner un algorithme de décision de la réductibilité inductive dans le cas général. Ces grammaires de formes normales sont aussi utilisées dans d'autres applications.

Si $t \in T(F, X)$, nous noterons NF_t l'ensemble $NF \cap \{t\sigma \,|\sigma \in \Sigma_g\}$. NF_t est ainsi l'ensemble des instances fermées irréductibles de t. Si $\underline{s} \in S$, $NF_{\underline{s}}$ est l'ensemble des termes fermés irréductibles qui sont de sorte $\underline{s}$. Enfin, si (t, d) est un terme contraint, $NF_{t,d}$ est l'ensemble des termes $t\sigma$ de NF_t tels que $T(F) \models d\sigma$. (Ou si l'on préfère, tels que σ soit une solution de d dans $T(F)$). On peut noter que, si x est une variable de sorte $\underline{s}$, $NF_x = NF_{\underline{s}}$. De plus, $NF = \bigcup_{\underline{s} \in S} NF_{\underline{s}}$.

Lorsque $t \equiv f(x_1, \ldots, x_n)$, NF_t sera aussi noté NF_f. Enfin, si f est un symbole fonctionnel n-aire et $A_1, \ldots A_n$ sont n ensembles de termes (de sortes voulues) $f(A_1, \ldots, A_n)$ désignera l'ensemble $\{f(t_1, \ldots, t_n) \mid \forall i, t_i \in A_i\}$.

Soit $C(LHS) = \cap_{g \in LHS} C(g)$ où $C(t)$ désigne l'ensemble des termes de $T(F)$ qui ne sont pas des instances de t. Si $f : \underline{s}_1 \times \ldots \times \underline{s}_n \rightarrow \underline{s}$, on obtient alors la relation:

$$NF_f = f(NF_{\underline{s}_1}, \ldots, NF_{\underline{s}_n}) \cap C(LHS)$$

Si l'on remarque que $NF = \bigcup_{f \in F} NF_f$, on obtient alors (lorsque S ne contient qu'un élement) :

$$NF = \bigcup_{f \in F} (f(NF, \ldots, NF) \cap C(LHS))$$

Qui n'est autre qu'une définition par point fixe de NF. Une telle relation doit permettre, à priori, de calculer une grammaire du langage NF. On pourrait écrire de semblables relations décrivant NF_t. Or un terme t est inductivement réductible si et seulement si NF_t est vide. L'intéret du calcul d'une grammaire engendrant NF (ou NF_t) est alors de pouvoir décider plus aisément de certaines propriétés du langage (et en particulier du vide, donc de la réductibilité inductive).

Le problème est d'exprimer $C(LHS)$ ainsi que l'intersection. Ce n'est, en fait, pas aisé car l'ensemble des termes fermés qui sont des instances d'un des termes de LHS est, en général, un langage d'arbres algébrique (ce qui peut être prouvé aisément), engendré par une grammaire IO [ES77]. Or le complémentaire d'un langage algébrique n'est pas

nécessairement algébrique. Il n'est donc pas possible d'espérer pouvoir calculer une grammaire algébrique de NF. C'est pourquoi nous introduirons une nouvelle espèce de grammaires: les *grammaires conditionnelles* qui peuvent s'apparenter, dans le cas des arbres, aux grammaires indexées de Aho [Aho68]. Néanmoins, dans le cas où tous les membres gauches de règles sont linéaires, le langage des termes fermés qui sont des instances de l'un d'eux est un langage d'arbres régulier (cf par exemple [GB85]). Comme la classe des langages réguliers est stable par complémentaire, NF est alors un langage régulier. Nos grammaires conditionnelles auront ainsi la propriété d'être des grammaires d'arbres régulières dans ce cas particulier. Notons que, même dans le cas linéaire, la grammaire que nous calculerons aura des propriétés particulières que nous utiliserons explicitement dans certaines applications: il n'est pas suffisant, même dans ce cas, d'effectuer les calculs classiques de déterminisation et de complémentation des automates.

5.2.1 Exemples de grammaires de formes normales

Avant de donner les définitions (techniques) de grammaire conditionnelle, dérivation, etc..., montrons sur deux exemples simples deux grammaires de formes normales et la façon dont elles sont obtenues.

Exemple 5.5 Ce premier exemple très simple est une grammaire de l'ensemble des formes normales fermées pour une spécification des entiers relatifs.

$$F = \{\ 0 :\rightarrow int;\ \ s, p : int \rightarrow int;\ \ + : int \times int \rightarrow int\ \}$$

$$\mathcal{R} = \{\ \begin{array}{ccc} s(p(x)) \rightarrow x & p(s(x)) \rightarrow x & 0 + x \rightarrow x \\ s(x) + y \rightarrow s(x + y) & p(x) + y \rightarrow p(x + y) & \end{array}\ \}$$

Pour calculer une grammaire engendrant NF_x, on commence par exprimer le fait qu'un terme fermé irréductible a pour racine l'un des symboles de F:

$$NF_x \ \rightarrow\ NF_0 \ \mid\ NF_{s(x)} \ \mid\ NF_{p(x)} \ \mid\ NF_{x_1+x_2}$$

Il nous faut maintenant donner des grammaires engendrant respectivement NF_0, $NF_{s(x)}$, $NF_{p(x)}$ et $NF_{x_1+x_2}$. 0 étant irréductible et n'ayant pas d'autre instance irréductible, nous pouvons écrire:

$$NF_0 \ \rightarrow\ 0$$

Pour calculer les règles associées à $NF_{s(x)}$, on cherche l'ensemble des termes fermés qui sont filtrés par $s(x)$ et qui ne sont pas filtrés par un membre gauche de règle : on résoud le problème de complément

$$\exists w, \forall y : \ x = w \wedge w \neq s(p(y))$$

Problème qui a 3 formes résolues:

- $x = 0$
- $\exists w_1 : \ x = s(w_1)$
- $\exists w_1, w_2 : \ x = w_1 + w_2$

Nous en déduisons les règles de grammaire:

$$NF_{s(x)} \rightarrow s(NF_0) \mid s(NF_{s(w_1)}) \mid s(NF_{w_1+w_2})$$

En effet, un terme est dans $NF_{s(x)}$ s'il n'est pas réductible à la racine (i.e. il n'est pas filtré par un membre gauche de règle) et si ses sous-termes stricts sont irréductibles.

De même, on obtient pour $NF_{p(x)}$:

$$NF_{p(x)} \rightarrow p(NF_0) \mid p(NF_{p(x)}) \mid p(NF_{x_1+x_2})$$

Enfin, en résolvant le problème

$$\exists w_1, w_2, \forall y_1, y_2 : x_1 = w_1 \wedge x_2 = w_2 \wedge w_1+w_2 \neq 0+y_1 \wedge w_1+w_2 \neq s(y_1)+y_2 \wedge w_1+w_2 \neq p(y_1)+y_2$$

on trouve une seule forme résolue ;

$$\exists w_1, w_2, w_3 : x_1 = w_1 + w_2 \wedge x_2 = w_3$$

nous obtenons ainsi finalement la grammaire

$$\begin{array}{lcl}
NF_x & \rightarrow & NF_0 \mid NF_{s(x)} \mid NF_{p(x)} \mid NF_{x_1+x_2} \\
NF_0 & \rightarrow & 0 \\
NF_{s(x)} & \rightarrow & s(NF_0) \mid s(NF_{s(x)}) \mid s(NF_{x_1+x_2}) \\
NF_{p(x)} & \rightarrow & p(NF_0) \mid p(NF_{p(x)}) \mid p(NF_{x_1+x_2}) \\
NF_{x_1+x_2} & \rightarrow & NF_{x_1+x_2} + NF_x
\end{array}$$

On peut maintenant noter que $NF_{x_1+x_2}$ est improductif puisque la seule dérivation possible de ce non-terminal le fait lui-même intervenir.Ce qui prouve que $x_1 + x_2$ est inductivement réductible. Par "nettoyage", on obtient alors la grammaire:

$$\begin{array}{lcl}
NF_x & \rightarrow & NF_0 \mid NF_{s(x)} \mid NF_{p(x)} \\
NF_0 & \rightarrow & 0 \\
NF_{s(x)} & \rightarrow & s(NF_0) \mid s(NF_{s(x)}) \\
NF_{p(x)} & \rightarrow & p(NF_0) \mid p(NF_{p(x)})
\end{array}$$

Mais les choses ne se passent pas toujours aussi facilement que dans l'exemple 5.5. Ce deuxième exemple introduit quelques-uns des problèmes qui se posent en général.

Exemple 5.6 Nous donnons ici une spécification des entiers modulo 2 avec l'addition. La différence essentielle avec l'exemple précédent est qu'il y a dans $\mathcal{R}$ un membre gauche de règle qui n'est pas linéaire.

$$F = \{ \quad 0 :\rightarrow int2; \quad s : int2 \rightarrow int2; \quad + : int2 \times int2 \rightarrow int2 \quad \}$$

$$\begin{array}{ll}
\mathcal{R} = \{ & s(s(0)) \rightarrow 0 \quad x + x \rightarrow 0 \\
& 0 + x \rightarrow x \quad x + 0 \rightarrow x \quad \}
\end{array}$$

Procédons de la même manière que dans l'exemple précédent. On commence par les règles:

$$\begin{array}{lcl}
NF_x & \rightarrow & NF_0 \mid NF_{s(x)} \mid NF_{x_1+x_2} \\
NF_0 & \rightarrow & 0
\end{array}$$

Puis, en résolvant le problème

$$\exists w : x = w \wedge s(w) \neq s(s(0))$$

on obtient les formes rèsolues:

- $x = 0$
- $\exists w : x = s(s(w))$
- $\exists w_1, w_2 : x = w_1 + w_2$
- $\exists w_1, w_2 : x = s(w_1 + w_2)$

d'où les règles:

$$NF_{s(x)} \rightarrow s(NF_0) \mid s(NF_{x_1+x_2}) \mid s(NF_{s(s(x))}) \mid s(NF_{s(x_1+x_2)})$$

Nous voyons déjà apparaitre ici une difficulté : nous avons introduits les nouveaux non-terminaux $NF_{s(s(x))}$ et $NF_{s(x_1+x_2)}$ dans ces règles. Il nous faudra rèsoudre à nouveau des problèmes de compléments pour ces non-terminaux.

En résolvant le problème

$$\exists w_1, w_2, \forall y : x_1 = w_1 \wedge x_2 = w_2 \wedge w_1 + w_2 \neq y + y \wedge w_1 + w_2 \neq y + 0 \wedge w_1 + w_2 \neq 0 + y$$

on obtient les formes résolues:

- $\exists w_1, w_2 : x_1 = s(w_1) \wedge x_2 = s(w_2) \wedge w_1 \neq w_2$
- $\exists w_1, w_2, w_3 : x_1 = w_1 + w_2 \wedge x_2 = s(w_3)$
- $\exists w_1, w_2, w_3 : x_1 = s(w_1) \wedge x_2 = w_2 + w_3$
- $\exists w_1, w_2, w_3, w_4 : x_1 = w_1 + w_2 \wedge x_2 = w_3 + w_4 \wedge w_1 \neq w_3$
- $\exists w_1, w_2, w_3, w_4 : x_1 = w_1 + w_2 \wedge x_2 = w_3 + w_4 \wedge w_2 \neq w_4$

On constate que certaines diséquations persistent dans ces formes résolues et qu'il n'est donc pas possible d'obtenir des règles ayant une forme aussi agréable que les précédentes. Nous écrirons:

$$\begin{array}{rcll} NF_{x_1+x_2} & \rightarrow & NF_{s(x_1)} + NF_{s(x_2)} & \text{Si } x_1 \neq x_2 \\ & \mid & NF_{x_1+x_2} + NF_{s(x)} & \\ & \mid & NF_{s(x)} + NF_{x_1+x_2} & \\ & \mid & NF_{x_1+x_2} + NF_{x_3+x_4} & \text{Si } x_1 \neq x_3 \\ & \mid & NF_{x_1+x_2} + NF_{x_3+x_4} & \text{Si } x_2 \neq x_4 \end{array}$$

On calcule maintenant les règles de grammaires associées à $NF_{s(s(x))}$ et à $NF_{s(x_1+x_2}$ de la même façon:

$$\begin{array}{rcl} NF_{s(s(x))} & \rightarrow & s(NF_{s(s(x))}) \\ & \mid & s(NF_{s(x_1+x_2)}) \\ NF_{s(x_1+x_2)} & \rightarrow & s(NF_{x_1+x_2}) \end{array}$$

On obtient alors un ensemble de règles dans lequel tous les non-terminaux introduits sont eux-mêmes définis. Nous verrons plus loin qu'une telle méthode termine bien dans tous les cas.

5.2.2 Grammaires conditionnelles

Définition 5.8 *Une* grammaire conditionnelle *est un quadruplet* $\mathcal{G} = (F, NT, A, P)$ *formé*

- *d'un ensemble fini* F *de symboles fonctionnels avec leur arité* τ *appelés* symboles terminaux
- *un ensemble fini de* non-terminaux NT *disjoints de* F. τ *est étendue à* NT. *De plus, chaque symbole de* NT *est associé à un terme contraint* (t, d) *construit sur* F *et* X. *On note* $N_{t,d}$, $N'_{t,d}$, $NF_{t,d}$, $NT_{t,d}$,... *les non-terminaux associés à* (t, d).
- *un* axiome $A \in NT$
- *un ensemble fini de* règles de production P *de l'une des formes*

$$N_{t,d}(X_1, \ldots, X_k) \rightarrow v[N_{t_1,d_1}(\vec{U_1}), \ldots, N_{t_n,d_n}(\vec{U_n})] \quad si \quad C$$

où

  - $X_1, \ldots, X_k \in X$ *et* $N_{t,d}$ *est d'arité* k
  - *pour tout* i, *il existe un renommage* θ_i *des variables de* t_i *tel que* $N_{(t_i,d_i)\theta_i} \in NT$
  - *pour tout* i, $\vec{U_i}$ *est une séquence de termes de* $T(F \cup NT, \{X_1, \ldots, X_k\})$
  - $v \in T(F)$[9]
  - $t_1, \ldots, t_n, t$ *sont sans variables communes.*
  - C *est une conjonctions de diséquations dont les variables sont contenues dans* $Var(t_1, \ldots, t_n)$.

ou bien $NF_{t,d}(X_1, \ldots, X_k) \rightarrow \lambda$

Lorsque l'ensemble d est vide, on note t à la place de $(t, \emptyset)$. Notons que, si chaque terme contraint de NT est une variable (ces variables sont alors de sortes différentes pour satisfaire la condition de non-équivalence par renommage) et si aucune règle ne contient de condition, on retrouve alors une grammaire algébrique d'arbres.

Définissons maintenant ce qu'est une dérivation dans une grammaire conditionnelle:

Définition 5.9 *Soient* U, U' *deux termes de* $T(F \cup NT, X)$ *et* C, C' *deux ensembles de diséquations dont les variables sont contenues respectivement dans* $Var(U)$ *et* $Var(U')$. *On dit que* (U, C) se dérive en (U', C') *dans la grammaire* $\mathcal{G}$ *(ce que l' on note* $(U, C) \Rightarrow_{\mathcal{G}} (U', C')$*) s'il existe une position* p *de* U *et une règle de production*

$$N_{t_0,d_0}(X_1, \ldots, X_k) \rightarrow v[N_{t_1,d_1}(\vec{U_1}), \ldots, N_{t_n,u_n}(\vec{U_n})] \quad si \quad C_0$$

dans $\mathcal{G}$ *tels que*

- $U/p \equiv N_{t,d}(u_1, \ldots, u_k)$ *avec* $u_1, \ldots, u_k \in T(F)$
- *Il existe un renommage* θ *tel que* $t_0\theta \equiv t$ *et* $d_0\theta \equiv d$.
- $U' \equiv U[v[N_{t_1,d_1}(\vec{U_1}, \ldots, N_{t_n,d_n}(\vec{U_n})]\{X_1 \rightarrow u_1; \ldots; X_k \rightarrow u_k\}\theta]_p$

[9] $v[NF_{t_1,d_1}(\vec{U_1}), \ldots, NF_{t_n,d_n}(\vec{U_n})]$ peut être de la forme $NF_{t_1,d_1}(\vec{U_1})$ si le remplacement a lieu à la racine.

- *C' est une forme résolue*[10] *distincte de $\perp$ du problème*

$$\exists Var(t_0) : c \wedge C_0\{X_1 \to u_1; \ldots; X_k \to u_k\} \wedge t_0 = v[t_1, \ldots, t_n]\{X_1 \to u_1; \ldots; X_k \to u_k\}$$

On suppose dans cette définition que, si $(U, C) \Rightarrow_{\mathcal{G}} (U', C')$, les non terminaux introduits dans U' ont des variables distinctes de celles qui apparaissaient dans U. (i.e., à chaque utilisation d'une règle de production, il faut utiliser un renommage des non-terminaux).

Notons que, lorsque la grammaire $\mathcal{G}$ est une grammaire algébrique d'arbres, la notion de dérivation que nous venons de définir coincide avec celle de dérivation IO ([ES77]). La condition $u_1, \ldots, u_k \in T(F)$ exprimant que les dérivations se font "à l'intérieur d'abord".

Le *langage engendré* par $S \in NT$ dans une grammaire conditionnelle $\mathcal{G} = (F, NT, A, P)$ (resp. le *langage engendré par* $\mathcal{G}$) est l'ensemble des termes $t \in T(F)$ tels qu'il existe des termes $t_1, \ldots, t_n \in T(F)$ (où n est l'arité de S, resp. l'arité de A) tels que $S(t_1, \ldots, t_n) \Rightarrow^*_{\mathcal{G}} t$ (resp. $A(t_1, \ldots, t_n) \Rightarrow^*_{\mathcal{G}} t$). On note $L(S, \mathcal{G})$ (resp. $L(\mathcal{G})$) le langage engendré par S dans $\mathcal{G}$ (resp. le langage engendré par $\mathcal{G}$).

Donnons quelques exemples de grammaires et de dérivations.

Exemple 5.7 Les exemples 5.5 et 5.6 sont des exemples de grammaires conditionnelles. Pour l'exemple 5.5, la séquence

$$NF_x \Rightarrow NF_{s(x)} \Rightarrow s(NF_{s(x)}) \Rightarrow s(s(NF_0)) \Rightarrow s(s(0))$$

est une dérivation pour la grammaire $\mathcal{G}$.

Donnons maintenant un exemple plus complexe.

Exemple 5.8 Dans cet exemple, la grammaire $\mathcal{G}$ décrit le langage NF_t dans le cas où t n'est pas linéaire et le système de réécriture n'est pas non plus linéaire. Nous voyons ici la nécéssité de toutes les constructions introduites dans la définition.

$$F = \{ \quad f : \underline{s} \times \underline{s} \to \underline{s}; \quad a :\to \underline{s} \quad \}$$

$$\mathcal{R} = \{ \quad f(x, x) \to x \quad \}$$

L'ensemble des termes fermés irréductibles qui sont des instances de $t = f(f(x, y), f(z, z))$ est décrit par la grammaire[11]

$$\mathcal{G} = (F, \{NF_t, NT_{\widehat{t}}, NF_{f(x_1,x_2)}, NF_{f(x_1,x_2),x_1 \neq x_2}, NF_a, NF_x\}, NF_t, P)$$

avec P:

$$\begin{array}{lcl}
NF_t & \to & NT_{f(x_1,f(x_2,x_3))}(NF_{f(w_1,w_2),w_1 \neq w_2}, NF_x) \\
NT_{f(x_1,f(x_2,x_3))}(X_1, X_2) & \to & f(X_1, f(X_2, X_2)) \\
NF_{f(x_1,x_2),x_1 \neq x_2} & \to & f(NF_{w_1}, NF_{w_2}) \quad \text{si} \quad w_1 \neq w_2 \\
NF_x & \to & NF_a \quad | \quad NF_{f(x_1,x_2)} \\
NF_a & \to & a \\
NF_{f(x_1,x_2)} & \to & f(NF_{w_1}, NF_{w_2}) \quad \text{si} \quad w_1 \neq w_2
\end{array}$$

[10] Il s'agit ici de forme résolue pour le système de règles de la section 3.5

[11] Cette grammaire peut être obtenue par un procédé analogue à celui qui est esquissé dans les exemple 5.5, 5.6.

La non linéarité de t rend nécessaire la deuxième règle de grammaire. La non-linéarité du système de réécriture entraine l'introduction de termes contraints et de conditions dans les règles.

La séquence suivante montre une dérivation pour cette grammaire:

$$\begin{array}{rcl} NF_t & \Rightarrow & NT_{f(x_1,f(x_2,x_3))}(NF_{f(w_1,w_2),w_1\neq w_2}, NF_x) \\ & \Rightarrow & NT_{f(x_1,f(x_2,x_3))}(NF_{f(w_1,w_2),w_1\neq w_2}, NF_a) \\ & \Rightarrow & NT_{f(x_1,f(x_2,x_3))}(NF_{f(w_1,w_2),w_1\neq w_2}, a) \\ & \Rightarrow & (\ NT_{f(x_1,f(x_2,x_3))}(f(NF_{w_1}, NF_{w_2}), a),\ w_1 \neq w_2) \\ & \Rightarrow & (\ NT_{f(x_1,f(x_2,x_3))}(f(NF_a, NF_{w_2}), a),\ a \neq w_2) \end{array}$$

puisque la seule forme résolue de $\exists w_1 : w_1 \neq w_2 \wedge w_1 = a$ est $w_2 \neq a$.

$$\Rightarrow \quad (\ NT_{f(x_1,f(x_2,x_3))}(f(a, NF_{w_2}), a),\ a \neq w_2)$$

Il n'y a plus ici qu'une seule dérivation possible:

- on ne peut utiliser la règle $NT_{f(x_1,f(x_2,x_3))}(X_1, X_2) \rightarrow f(X_1, f(X_2, X_2))$ car il faudrait d'abord avoir dérivé NF_{w_2}. (c'est la condition "IO", ou, dans notre définition, la condition $u_1, \ldots, u_k \in T(F)$).
- On ne peut appliquer la règle $NF_{w_2} \rightarrow NF_a$ car $\exists w_2 : w_2 \neq a \wedge w_2 = a \mapsto^* \perp$.

Il ne reste alors que la règle $NF_{w_2} \rightarrow NF_{f(w_3,w_4)}$ qui conduit à:

$$NF_t \Rightarrow^*_{\mathcal{G}} NT_{f(x_1,f(x_2,x_3))}(f(a, NF_{f(w_3,w_4)}), a)$$

On peut remarquer qu'en fait, sur cet exemple, le langage engendré est vide : on ne peut jamais atteindre de terme de $T(F)$.

5.2.3 Le point de vue du "reconnaisseur"

De façon classique, les règles de grammaires peuvent aussi être vues comme des règles de réduction (c'est le point de vue "reconnaisseur").

Définition 5.10 *Soient (t, e) et (t', e') deux termes de $T(F \cup NT)$ auxquels on a adjoint des systèmes d'équations e et e' dont les variables sont celles des indices des non-terminaux apparaissant dans t et t' respectivement. On dit que (t, e)* se réduit en *(t', e') par la grammaire conditionnelle $\mathcal{G}$ (ce que l'on note $(t, e) \rightarrow_{\mathcal{G}} (t', e')$) s'il existe une règle*

$$N_{t_0,d_0}(X_1, \ldots, X_k) \rightarrow v[N_{t_1,d_1}(\vec{U_1}), \ldots, N_{t_n,d_n}(\vec{U_n})] \quad si \quad \delta_0$$

de $\mathcal{G}$ et une substitution σ de domaine $\{X_1, \ldots X_k\}$ telles que:

- $t/p \equiv v[N_{t_1,d_1}(\vec{U_1}), \ldots, N_{t_n,d_n}(\vec{U_n})]\sigma$
- $t' \equiv t[N_{t_0,d_0}(X_1\sigma, \ldots, X_k\sigma)]_p$
- *e' est une forme résolue distincte de $\perp$ de*

$$\exists Var(t_1, \ldots, t_n) : t_0 = v[t_1, \ldots, t_n]\sigma \wedge \delta_0\sigma \wedge e$$

On suppose de plus que les variables de t_0 ont été renommées de façon à ne pas apparaitre dans e.

Remarquons que e' est bien une conjonction d'équations, les variables de $C_0\sigma$ étant quantifiées existentiellement.

$t \in L(\mathcal{G})$ ssi il existe $t_1, \ldots, t_n \in T(F)$ tels que $t \to_{\mathcal{G}}^* A(t_1, \ldots, t_n)$. Ce point de vue est souvent plus simple que le point de vue génératif car les termes impliqués dans les réductions sont toujours des termes fermés. (Ce qui n'est pas le cas des termes impliqués dans les dérivations).

Si R est une règle de $\mathcal{G}$, on note encore $(t, e) \to_R (t', e')$ si $(t, e) \to_{\mathcal{G}} (t', e')$ et que la réduction est effectuée en utilisant la règle R.

Exemple 5.9 Reprenons l'exemple 5.7. On peut construire la chaine de réductions suivante :

$$\begin{array}{lll} f(a, f(a,a)) & \to_{\mathcal{G}} & f(NF_a, f(a,a)) \\ & \to_{\mathcal{G}}^* & (f(NF_{x_1}, f(NF_{x_2}, NF_{x_3})), x_1 = a \wedge x_2 = a \wedge x_3 = a) \end{array}$$

Aucune règle n'est plus applicable car les seules qui le seraient éventuellement conduisent à un système d'équations e' égal à $\bot$.

Comme les systèmes d'équations associés dans ces réductions à un terme sont toujours en forme résolue, on utilisera plutôt une substitution pour désigner ce système d'équations. Si bien que la relation $\to_{\mathcal{G}}$ liera des termes munis d'une substitution.

5.2.4 Grammaires de formes normales

On supposera fixé pour tout terme t un ensemble complet de positions de t. Nous omettrons ainsi les références à cet ensemble complet de positions, tout en en utilisant les propriétés.

$IR(t, d)$ est défini par:

- si t n'est pas une variable, $IR(t, d)$ est l'ensemble (fini, d'après la section 4.6) des formes irréductibles du problème de complément $\mathcal{C}((t, d), LHS)$
- si t est une variable de sorte $\underline{s}$, $IR(t, d)$ est l'ensemble des problèmes $\exists w_1, \ldots, w_n$: $t = f(w_1, \ldots, w_n)$ pour $f \in F$ dont le codomaine est $\underline{s}$.

D'après le théorème 4.66, les problèmes de $IR(t, d)$ sont de la forme:

$$\exists \vec{w} : x_1 = t_1 \wedge d_1 \wedge \ldots x_n = t_n \wedge d_n \wedge \delta$$

où $x_1, \ldots, x_n$ sont les variables de $\hat{t}$, $t_1, \ldots, t_n$ sont des termes lineaires et sans variables partagées, $(t_1, d_1), \ldots, (t_n, d_n)$ sont des termes contraints, ...

Un sous-ensemble A de $IR(t, d)$ est dit *complet* si $NF_{t,d}$ est la réunion des ensembles

$$\{\hat{t}\sigma \mid \forall i, x_i\sigma \in NF_{t_i,d_i} \text{ and } \sigma \in \mathcal{S}(\delta, T(F))\}$$

pour $\exists \vec{w} : x_1 = t_1 \wedge d_1 \wedge \ldots x_n = t_n \wedge d_n \wedge \delta \in A$.

Proposition 5.11 *Pour tout terme contraint (t, d), $IR(t, d)$ est complet.*

Ce résultat (qui est une conséquence immédiate des définitions et du théorème 4.66) permet d'associer à chaque langage $NF_{t,d}$ des règles de grammaire. Nous confondrons dans cette notation le non terminal $NF_{t,d}$ (d'arité 0) et le langage qu'il engendre. $NT_{\widehat{t}}$ désignera un nouveau non terminal (d'arité égale au nombre de variables de $\widehat{t}$). Nous appelerons alors règles de grammaire (resp. *ensemble complet de règles de grammaire* relatif à A) *associées à* (t, d), les règles:

$$NF_{t,d} \rightarrow NT_{\widehat{t}}(NF_{t_1,d_1}, \ldots, NF_{t_n,d_n}) \quad \text{si} \quad \delta$$

où $x_1 = t_1 \wedge d_1 \wedge \ldots \wedge x_n = t_n \wedge d_n \wedge \delta$ est un problème de $IR(t, d)$ (resp. est un problème de A, sous-ensemble complet de $IR(t, d)$) ainsi que la règle:

$$NT_{\widehat{t}}(X_1, \ldots, X_n) \rightarrow \widehat{t}\{x_1 \rightarrow X_1, \ldots, x_n \rightarrow X_n\}$$

si $x_1, \ldots, x_n$ sont les variables de $\widehat{t}$.

Si $IR(t, d)$ (resp. A) ne contient que $\bot$, l'ensemble complet de règles de grammaire associé à (t, d) est constitué de la seule règle $NF_{t,d} \rightarrow \lambda$.

En fait ces règles de "second niveau" ne sont nécessaires que lorsque $\widehat{t}$ n'est pas linéaire[12]. Si bien que, dans le cas où t est linéaire, la règle faisant intervenir $NF_{\widehat{t}}$ est combinée avec les autres pour donner les règles suivantes (encore appelées règles de grammaire associées à (t, d), resp. ensemble complet de règles de grammaire associées à (t, d)):

$$NF_{t,d} \rightarrow \widehat{t}\{x_1 \rightarrow NF_{t_1,d_1}, \ldots, x_n \rightarrow NF_{t_n,d_n}\} \quad \text{si} \quad \delta$$

Comme, de plus, les grammaires que nous considérerons contiendront au plus un non terminal $NF_{t,d}$ dans lequel t n'est pas linéaire, nous appellerons

Définition 5.12 *Une* présentation de formes normales *(PFN en abrégé) est une grammaire conditionnelle (F, NT, A, P) dans laquelle NT est un ensemble constitué -de non-terminaux de la forme $NF_{t,d}$ (d'arité 0) où t est un terme linéaire -de l'axiome NF_{t_0,d_0} (d'arité 0) -éventuellement du non-terminal $NT_{\widehat{t_0}}$. P est un ensemble fini d'ensembles complets de règles de grammaires associées à des termes contraints (t, d) tels que $NF_{t,d} \in NT$.*

Dans une présentation, les règles de production sont toujours de la forme *membre gauche* $\rightarrow$ *membre droit* si *condition.* Cette condition étant éventuellement vide.

D'après cette définition, les présentations de formes normales forment une sous-classe stricte des grammaires conditionnelles. En particulier, on n'autorise seulement deux non-terminaux au plus à être associés à un même terme (contraint) à renommage près.

Définition 5.13 *Une* grammaire de formes normales *(GFN en abrégé) est une présentation de formes normales dans laquelle tout non-terminal apparaissant en membre droit de règle de production apparait aussi en membre gauche.*

[12] C'est à dire lorsque le langage des instances de t est algébrique mais n'est pas rationnel

Les exemples 5.5,5.6 et 5.7 sont des exemples de grammaires de formes normales.

D'autre part, de la définition d'un ensemble complet de règles de grammaire relatif à (t, d) il résulte:

Proposition 5.14 *Le langage engendré par une GFN* $(NT, F, NF_{t,d}, P)$ *est* $NF_{t,d}$.

Cette proposition autorise la confusion (que nous avons déjà faite et continuerons à faire) entre le non-terminal $NF_{t,d}$ et le langage de formes normales correspondant.

Il ne reste plus qu'à montrer comment calculer une grammaire de formes normales pour un terme contraint (t, d) donné.

5.2.5 Construction des grammaires de formes normales

Soit $\Pi_0 = (F, NT_0, A, P_0)$ une présentation de formes normales. Considérons l'algorithme:

Complete(F, NT, A, P) =
  Si (F, NT, A, P) est une GNF alors (F, NT, A, P)
  Sinon
    Soit $NF_{t,d}$ un non terminal apparaissant dans un membre droit et pas dans un membre gauche de règle de P.
    Soit P_1 l'ensemble des règles de grammaire associées à (t, d).
    Soit NT_1 l'ensemble des non-terminaux apparaissant dans P_1 et pas dans NT.
    Complete$(F, NT \cup NT_1, A, P \cup P_1)$

Théoreme 5.15 Complete *termine lorsqu'appliqué à une présentation de formes normales* Π_0 *d'axiome* $NF_{t,d}$. *La grammaire de formes normales* $\mathcal{G}$ *ainsi obtenue engendre le langage* $NF_{t,d}$.

Preuve

D'après le théorème 4.66, les formes irréductibles de problèmes de compléments sont des problèmes simples (cf définition 4.55. La propriété 4 des problèmes simples (linéarité des solutions) entraine en particulier que tous les non-terminaux ajoutés au cours de la complétion sont de la forme $NF_{t,d}$ où t est un terme linéaire. La propriété 8 des problèmes simples entraine que l'assertion:

$$\forall NF_{t,d} \in NT,\ h(t) \leq \max(\ \max\{h(u) | NF_{u,d'} \in NT_0\}\ ,\ \max\{h(l) | l \in LHS\}\ )$$

est invariante par application de Complete. Cette propriété entraine que NT reste contenu dans l'ensemble fini des termes contraints (t, d) tels que t a une profondeur inférieure à n fixé[13]. Il en résulte que, après un nombre fini d'applications de Complete, tout non-terminal ayant une occurrence en membre droit de règle a une occurrence en membre gauche de règle.

La propriété d'arité résulte quant à elle du fait qu'on ne rajoute que des termes containts linéaires, comme remarqué ci-dessus. □

[13] Notons que, comme d est une conjonction de diséquations liant des sous-termes de t, pour un t fixé il n'y a qu'un nombre fini de contraintes possibles.

Les exemples 5.5 et 5.6 montrent comment cette "complétion" des présentations est effectuée. Lorsqu'on souhaite obtenir une grammaire engendrant NF_t il suffit ainsi de compléter la présentation obtenue à l'aide d'un ensemble complet de règles de grammaires associées à t. Présentation qui n'est qu'une traduction de $IR(t)$. Quand nous parlerons de *la* GNF engendrant NF_t nous désignerons celle qui est obtenue de cette façon.

5.2.6 Quelques propriétés simples des grammaires de formes normales

La première de ces propriétés élémentaires a été annoncée en introduction:

Proposition 5.16 *Si tous les termes de LHS sont linéaires et t est un terme linéaire, la GNF engendrant NF_t est une grammaire d'arbre rationnelle.*

Preuve
Remarquons que, d'après la propriété 7 des problèmes simples et le théorème 4.66, aucun problème de $IR(t)$ ne comporte de diséquations lorsque tout terme de LHS est linéaire. Il en résulte que l'absence de condition dans les règles de production est un invariant de Complete.

D'autre part, la linéarité de t entraine qu'aucun non-terminal n'est d'arité supérieure à 0, puisque, comme nous l'avons déjà vu, si $NF_{u,d}$ est ajouté par complétion, u est linéaire
□

Dans le cas général, étant donné un terme contraint (t, d), il existe toujours une grammaire de formes normales ayant certaines propriétés particulières et qui engendre $NF_{t,d}$:

Comme dans tous les formalismes grammaticaux, tout langage admet des grammaires qui sont plus "simples" que les autres:

Définition 5.17 *Une grammaire de formes normales est* réduite *si tout non-terminal de cette grammaire engendre un langage non vide.*

Proposition 5.18 *Un terme t n'est pas inductivement réductible ssi il existe une grammaire réduite qui l'engendre.*

Cette proposition montre la démarche à suivre pour les preuves de réductibilité inductive: 1) calculer une grammaire de formes normales de NF_t (par exemple en complétant la présentation correspondant à $IR(t)$) 2) Éliminer les improductifs de cette grammaire. (On appelle *improductif* un non-terminal engendrant un langage vide). Nous avons vu comment effectuer la première étape. L'objet de la section suivante est de montrer comment effectuer la seconde.

5.3 Nettoyage des grammaires de formes normales

Dans le cas des grammaires d'arbre régulières (ou des grammaires de mots hors contexte) l'algorithme de réduction est simple. On peut le décrire ainsi:
$\mathcal{G} = (F, NT, A, P)$ est donnée.

1. Marquer tous les non-terminaux N tels qu'il existe une arbre terminal t vérifiant $t \rightarrow_{\mathcal{G}} N$. **Marque** prend la valeur "vrai" si l'on a marqué ainsi au moins un non-terminal et "faux" sinon. Soit NT_0 l'ensemble des non-terminaux marqués.

2. Tant que **Marque**

   - Marquer les non-terminaux $N \in NT - NT_0$ tels qu'il existe un arbre $t \in T(F, NT_0)$ vérifiant $t \rightarrow_{\mathcal{G}} N$
   - Si aucun non-terminal n'a été marqué à l'étape ci-dessus, alors **Marque** prend la valeur "faux".

3. Si l'axiome A n'est pas marqué alors le langage engendré est vide. Sinon, soit $\mathcal{G}'$ la grammaire (F, NT_0, A, P_0) où P_0 est le sous-ensemble des règles de P qui ne contiennent que des arbres de $T(F, NT_0)$. $\mathcal{G}'$ est une grammaire réduite engendrant A.

On peut aussi résumer cet algorithme en disant : on calcule $\rightarrow_{\mathcal{G}}^{n}$ où n est le cardinal de NT et l'on élimine de $\mathcal{G}$ tous les non-terminaux qui ne sont pas dans le graphe de cette relation.

Dans tous les cas l'algorithme termine car il y a au plus $n = |NT|$ passages dans la boucle. Il est correct car les non-terminaux marqués engendrent un langage non vide.

Nous allons utiliser un algorithme semblable pour nettoyer les grammaires de formes normales. Mais $\rightarrow_{\mathcal{G}}$ est une réduction conditionnelle dans notre cas. Nous ne pourrons pas non plus nous contenter de "marquer" les non terminaux engendrant un langage non vide. Il nous faudra aussi conserver les éléments du langage qu'ils engendrent afin de pouvoir tester les conditions par la suite. De même, nous ne pourrons pas nous contenter d'un seul élément du langage engendré, car, même si cet élément ne vérifie pas les conditions d'une réduction, il se peut que d'autres éléments du même langage les satisfassent. Les suites de réductions que nous envisagerons pourront ainsi, au contraire du cas des langages rationnels, utiliser plusieurs fois une même règle de grammaire. Mais le nombre d'utilisations d'une même règle restera borné car, informellement, les conditions (qui sont des systèmes de diséquations) admettent des solutions dès que l'on autorise les variables qui les composent à prendre suffisamment de valeurs distinctes.

Ce problème de nettoyage de grammaires conditionnelles peut aussi être rapproché du problème de l'accessibilité dans les réseaux de Pétri à file [Fin86]. Si l'on considère en effet les règles de réduction comme des règles d'un système de transition dans lequel les états sont les ensembles finis de termes fermés irréductibles. Le calcul de l'ensemble des termes qui se réduisent en n étapes en un non-terminal donné (que nous appelerons calcul de la présentation de formes normales) peut aussi être rapproché du calcul d'arbres de couverture de systèmes de transitions [Fin86].

On peut donc se demander s'il n'est pas possible d'utiliser des résultats existants sur les réseaux de Pétri. Pour utiliser les résultats de [Fin86] (dont l'étude semble la plus proche de notre problème) il faudrait munir l'ensemble des ensembles de termes fermés irréductibles d'un ordre de façon à ce que le système obtenu soit "bien structuré" au sens de [Fin86]. L'arbre de couverture serait alors un arbre fini sur lequel l'accessibilité est un

problème trivial. Le problème essentiel est que, pour obtenir un système bien structuré, il faut pouvoir calculer les limites de suites croissantes d'états et surtout *pouvoir comparer* ces limites. C'est peut-être possible, mais semble en tous cas un problème suffisamment complexe : il faudrait développer une théorie des calculs sur les ensembles infinis d'arbres contraints. nous avons donc préféré ici une méthode directe. Néanmoins, cette direction de recherche peut mériter plus d'attention à l'avenir. Certains outils que nous emploierons s'inspireront d'ailleurs de ceux qui sont proposés dans [Fin86].

5.3.1 Etats d'un calcul d'une grammaire de formes normales

Dans la suite $\mathcal{G} = (F, NT, NF_{t_0,d_0}, P)$ est une grammaire de formes normales fixée. CT désignera l'ensemble des termes contraints qui sont associés à non-terminal de NT. Si bien que l'application $[\![\cdot]\!]$ qui associe à chaque terme contraint (t,d) de CT le langage $NF_{t,d}$ est une bijection de CT dans $NT - \{NT_{\widehat{t_0}}\}$.

Définition 5.19 *Un* état du calcul *de $\mathcal{G}$ est une application C de CT dans l'ensemble des parties de $T(F)$ telle que, pour tout $(t,d) \in CT$, $C(t,d) \subseteq [\![t,d]\!]$.*

Si l'on ajoute la contrainte que les états doivent être complets, c'est-à-dire (informellement) que chaque sous-terme d'un élément de $C(t,d)$ appartient à un certain $C(t',d')$, alors on obtient un ensemble d'états d'un système de transition qui est muni d'un bel ordre (au sens de [Fin86]).

C_0 désignera l'état particulier qui associe à tout élément de CT l'ensemble vide.

Définition 5.20 *Le* calcul *de $\mathcal{G}$ est l'application $\mathcal{F}$ qui associe à chaque état C de $\mathcal{G}$ l'état $\mathcal{F}(C)$ défini par:*

- *si t n'est pas une variable,*

$$(\mathcal{F}(C))(t,d) = C(t,d) \cup \{u \in [\![t,d]\!] \mid \forall p \in \widetilde{Q_t}, u/p \in \bigcup_{w \in CT} C(w)\}$$

- *si t est une variable de sorte $\underline{s}$,*

$$(\mathcal{F}(C))(t) = \bigcup_{(u,d) \in CT, sort(u) = \underline{s}} C(u,d)$$

Étant donné un état C du calcul de $\mathcal{G}$, il est aisé de calculer $\mathcal{F}(C)$. Appliquer $\mathcal{F}$ consiste en effet essentiellement à chercher toutes les réductions possibles (en un pas) à partir des termes contenus dans l'image de C. Plus formellement, notons D_C l'ensemble $\bigcup_{w \in CT} C(w)$, notons Σ_C l'ensemble des substitutions σ telles que $\forall x \in Dom(\sigma), x\sigma \in D_C$. Notons enfin (t,σ) les termes munis d'équations en forme résolue. Alors,

Définition 5.21 *Un état C* se réduit par la règle *R de $\mathcal{G}$ en un état C' (noté $C \rightarrow_R C'$) si*

- *R est de la forme $NF_{t,d} \rightarrow v[NF_{t_1,d_1}, \ldots, NF_{t_n,d_n}]$ Si δ*
- *C et C' ne diffèrent que sur (t,d)*

- $\exists \sigma \in \Sigma_C$, $\exists v \in T(F \cup NT)$, $\exists \sigma'$ *tels que*
  - *si* $v \in T(F)$, $(v, \sigma) \rightarrow_R (NF_{t,d}, \sigma')$
  - *si* $v \equiv NT_{\widehat{t_0}}$, $\exists \vec{U} \in T(F \cup NT)^k$,

$$(v, \sigma) \rightarrow_{\mathcal{G}} (NT_{\widehat{t_0}}(\vec{U}), \sigma) \rightarrow_R (NF_{t_0,d_0}, \sigma')$$

- $C'(t,d) = C(t,d) \cup \{t\sigma'\}$

On définit la réduction des états dans la grammaire $\mathcal{G}$ par $C \rightarrow_{\mathcal{G}} C'$ ssi il existe une règle R de $\mathcal{G}$ telle que $C \rightarrow_R C'$.

La proposition suivante montre comment il est possible de calculer $\mathcal{F}(C)$ en utilisant $\rightarrow_{\mathcal{G}}$:

Proposition 5.22 *Pour tout état C du calcul de $\mathcal{G}$, pour tout $(t,d) \in CT$*

$$\mathcal{F}(C)(t,d) - C(t,d) = \bigcup_{C \rightarrow_{\mathcal{G}} C'} C'(t,d)$$

Cette proposition est une conséquence directe des définitions (de $IR(t,d)$, des règles associées à (t,d), de $\rightarrow_R$).

On peut aussi remarquer que, pour le calcul des éléments de $\mathcal{F}(\mathcal{F}(C))$, il n'est pas nécessaire de considérer tous les éléments de $\mathcal{F}(C)$; on peut utiliser l'identité:

$$\mathcal{F}(\mathcal{F}(C)) = \mathcal{F}(C \cup (\mathcal{F}(C) - C)) = \mathcal{F}(C) \cup \mathcal{F}(\mathcal{F}(C) - C) = C \cup \mathcal{F}(\mathcal{F}(C) - C)$$

et donc ne calculer $\mathcal{F}$ que sur $\mathcal{F}(C) - C$.

Nous verrons aussi dans la suite qu'il n'est pas nécessaire de calculer tous les éléments des ensembles $C(t,d)$ mais seulement "un nombre suffisant"; nous allons montrer qu'il existe un entier n tel que $\mathcal{F}^n(C_0)(t,d) = \emptyset$ entraine, pour tout p, $\mathcal{F}^p(C_0)(t,d) = \emptyset$. Plus précisément, nous allons montrer qu'il existe une fonction $\mathcal{F}'$ (qui ne prend qu'un nombre fini de valeurs) telle que si, pour tout $(t,d) \in CT$, $\mathcal{F}'(C)(t,d) = C(t,d)$, alors pour tout n et pour tout non-terminal (t,d), $\mathcal{F}'(\mathcal{F}^n(C))(t,d) = C(t,d)$.

L'algorithme de décision du vide de $[\![t,d]\!]$ pourra alors s'énoncer ainsi:

1. $C := C_0$
2. Tant que $\mathcal{F}'(\mathcal{F}(C)) \neq \mathcal{F}'(C)$ et que $C(t_0,d_0) = \emptyset$
   $C := \mathcal{F}(C)$
3. Si $C(t_0,d_0) \neq \emptyset$ alors $[\![t_0,d_0]\!]$ n'est pas vide sinon $[\![t_0,d_0]\!]$ est vide.

Avant de donner la construction (technique) de $\mathcal{F}'$, montrons sur un exemple son fonctionnement.

Exemple 5.10 $F = \{0, s, g, f\}$, 0 est une constante, s, g sont unaires et $+$ est binaire. $\mathcal{R}$ contient les règles:

$$\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl} s(s(0)) & \to & 0 & s(g(x)) & \to & x & g(s(x)) & \to & x \\ x+x & \to & 0 & x+0 & \to & x & 0+x & \to & x \\ x+g(y) & \to & g(x+y) & g(x)+y & \to & g(x+y) & & & \end{array}$$

Ce système de réécriture n'est pas confluent, mais cela n'a pas d'importance. $[\![x_1+x_2]\!]$ est vide, mais cela n'est pas immédiat, car il faut reconnaitre que les termes irréductibles construits sans $+$ sont soit de la forme $g^n(0)$ soit l'un des deux termes 0 ou $s(0)$. Ensuite il faut remarquer que tout terme de la forme $t_1 + t_2$ où t_1 et t_2 sont irréductibles et ne contiennent pas d'occurrence de $+$ est réductible. En effet, si t_1 ou t_2 a pour racine g, l'une des deux dernières règles de $\mathcal{R}$ s'applique. Si l'un des deux termes est 0 l'une des règles $x + 0 \to x$ ou $0 + x \to x$ s'applique. Enfin, si $t_1 \equiv t_2 \equiv s(0)$, la règle $x + x \to 0$ s'applique.

Cet exemple est un des plus compliqués qu'on puisse imaginer (et qui puisse s'écrire en quelques lignes) puisque $[\![x_1 + x_2]\!]$ est effectivement vide (s'il ne l'était pas, le problème serait rapidement résolu en exhibant un contre-exemple), l'un des membres gauche n'est pas linéaire et sa présence est nécessaire pour obtenir le résultat, enfin l'ensemble des termes fermés irréductibles est infini.

Le calcul de la grammaire de formes normales est, à peu de choses près, celui de l'exemple 5.6. Le résultat de cet algorithme (de calcul de la grammaire) est donné dans la figure 5.1.

Le tableau de la figure 5.2 illustre l'itération du calcul de $\mathcal{G}$. On trouve à l'intersection de la ligne i et de la colonne N l'ensemble $\mathcal{F}^i(C_0)(N) - \mathcal{F}^{i-1}(C_0)(N)$. Nous y avons aussi mentionné ce qu'on obtiendrait en joutant les règles de grammaire correspondant à NF_x et $NF_{g(x)}$:

$$\begin{array}{lcl@{\quad}l@{\quad}l@{\quad}l} NF_x & \to & NF_0 & NF_{s(x)} & NF_{g(x)} & NF_{x_1+x_2} \\ NF_{g(x)} & \to & g(NF_0) & g(NF_{g(x)} & g(NF_{x_1+x_2}) & \end{array}$$

Ces règles ne sont pas utiles pour la décision du vide de $[\![x_1+x_2]\!]$ mais permettent d'obtenir une grammaire de l'ensemble de tous les termes fermés irréductibles.

Revenons à la figure 5.2. On remarque que (si l'on excepte les eux dernières colonnes) la ligne 3 ne contient rien. Cela prouve que $\mathcal{F}^3(C_0) = \mathcal{F}^2(C_0)$ et donc que $\mathcal{F}'(\mathcal{F}(C)) = \mathcal{F}'(C)$ pour $C = \mathcal{F}^2(C_0)$. Nous sommes donc dans un cas d'arrêt de l'algorithme : sans poursuivre, il est possible d'affirmer que

$$[\![x_1 + x_2]\!] = [\![s(s(x))]\!] = [\![s(g(x))]\!] = [\![s(x_1 + x_2)]\!] = \emptyset$$

Autrement dit $x_1 + x_2$ est inductivement réductible. On remarquera que sur cet exemple, peu de termes ont été calculés.

Plus généralement, l'algorithme s'arretera -ou bien lorsqu'on rencontre une colonne vide -ou bien lorsque toutes les lignes sont non vides. Mais, en général, on ne se trouve pas dans un cas aussi simple que celui de l'exemple ci-dessus (c'est-à-dire un cas ou l'ensemble

$$
\begin{array}{lcl}
NF_{x_1+x_2} & \rightarrow & NF_{s(x_1)} + NF_{s(x_2)} \text{ si } x_1 \neq x_2 \\
 & | & NF_{s(x_1)} + NF_{x_2+x_3} \\
 & | & NF_{x_1+x_2} + NF_{s(x_3)} \\
 & | & NF_{x_1+x_2} + NF_{x_3+x_4} \text{ si } x_1 \neq x_3 \\
 & | & NF_{x_1+x_2} + NF_{x_3+x_4} \text{ si } x_2 \neq x_4 \\
NF_{s(x)} & \rightarrow & s(NF_0) \\
 & | & s(NF_{x_1+x_2}) \\
 & | & s(NF_{s(s(x))}) \\
 & | & s(NF_{s(x_1+x_2)}) \\
 & | & s(NF_{s(g(x))}) \\
NF_0 & \rightarrow & 0 \\
NF_{s(s(x))} & \rightarrow & s(NF_{s(s(x))}) \\
 & | & s(NF_{s(g(x))}) \\
 & | & s(NF_{s(x_1+x_2)}) \\
NF_{s(x_1+x_2)} & \rightarrow & s(NF_{x_1+x_2}) \\
NF_{s(g(x))} & \rightarrow & \lambda
\end{array}
$$

Figure 5.1: La grammaire de formes normales de l'exemple 5.10

i CT	0	$s(x)$	$x_1 + x_2$	$s(s(x))$	$s(x_1 + x_2)$	$s(g(x))$	$g(x)$	x
1	0							
2		$s(0)$					$g(0)$	0
3							$g(g(0))$	$s(0), g(0)$
4							$g(g(g(0)))$	$g(g(0))$

Figure 5.2: calcul de $\mathcal{F}^n$ et nettoyage de la grammaire de l'exemple 5.10

des formes normales "pertinentes" est fini). Comme nous l'avons mentionné ci-dessus, nous nous limiterons au calcul d'un nombre suffisant de termes dans chaque ligne (ce nombre étant fini, l'algorithme terminera alors trivialement). Montrons sur l'exemple comment l'on peut calculer cette borne théorique.

La figure 5.3 illustre le calcul des états de la grammaire de formes normales. Oublions tout d'abord les chiffres inscrits dans les rectangles ainsi que les chiffres indiqués entre crochets. Chaque non-terminal est représenté dans une éllipse. Chaque point désigne une transition possible (ou une règle de grammaire) les flèches aboutissant à un point figurent les termes nécessaires pour utiliser la réduction associée au point et la flèche qui part du point figure le terme qui est construit en utilisant cette réduction. Par exemple, de 0 part une flèche étiquetée par 1 et du point auquel elle aboutit part une flèche vers $s(x)$, elle aussi étiquetée par 1. Cela correspond à la règle $NF_{s(x)} \rightarrow s(NF_0)$. Pour chaque terme dans NF_0 il est en effet possible de calculer un terme dans $NF_{s(x)}$. Les chiffres étiquetant les flèches figurent le nombre de termes du langage correspondant qui sont soit produits par la réduction, soit nécessaires pour effectuer la réduction. Par exemple, la règle

$$NF_{x_1+x_2} \rightarrow NF_{s(x_1)} + NF_{s(x_2)} \quad \text{Si} \quad x_1 \neq x_2$$

ne peut conduire à une réduction que si $C(s(x))$ contient au moins deux termes a et b (flèche étiquetée par 2 et issue de $s(x)$). Mais dans ce cas, 2 termes sont produits dans $NF_{x_1+x_2}$: $a+b$ et $b+a$. (Ce qui correspond à l'étiquette 2 sur la flèche aboutissant à x_1+x_2).

On peut alors imaginer que, initialement, seule la boite 0 contient un élément et que $\mathcal{F}$ correspond à l'application de toutes les transitions possibles à chaque étape. Mais ici, au contraire des réseaux de Pétri, les boites ne perdent pas les termes qu'elles avaient: leur contenu ne fait que s'accoitre.

Venons en maintenant aux nombres entre crochets: ces nombres figurent le nombre de sous-termes différents aux différentes positions du terme considéré. En effet, considérons par exemple la règle

$$NF_{x_1+x_2} \rightarrow NF_{x_1+x_2} + NF_{x_3+x_4} \quad \text{Si} \quad x_1 \neq x_3$$

Une réduction par cette règle ne peut être effectuée que si l'on dispose de deux termes t_1, t_2 dans $NF_{x_1+x_2}$ tels que $t_1/1 \not\equiv t_2/1$. C'est le sens de l'étiquette $2[2,1]$ sur la flèche issue de x_1+x_2: 2 termes dans $NF_{x_1+x_2}$ sont nécessaires, ces deux termes ayant des sous-termes distincts à la position 1.

Voyons maintenant comment s'effectue le calcul de la borne au nombre de termes à calculer dans chaque boite: si l'on s'intéresse, par exemple, au vide de $[\![x_1+x_2]\!]$, il suffit de calculer un terme de ce langage (nombre inscrit dans le rectangle attaché à cette boite sur la figure 5.3). Puis, considérant les transitions qui permettent d'arriver à cette boite, on voit que 2 termes dans $[\![s(x)]\!]$ suffisent à en obtenir 1 dans $[\![x_1+x_2]\!]$. On inscrit donc le nombre 2 dans le rectangle associé à la boite $s(x)$. Et ainsi de suite... Pour chaque boite, il suffit de garder le nombre minimum, si l'on obtient plusieurs nombres par des chemins différents.

On voit que, sur cet exemple, il pouvait être calculé d'avance qu'au plus deux termes étaient utiles dans chaque ligne. c'est-à-dire que $|\mathcal{F}'(C)(N)| \leq 2$ pour tout $N \in CT$ et tout état C.

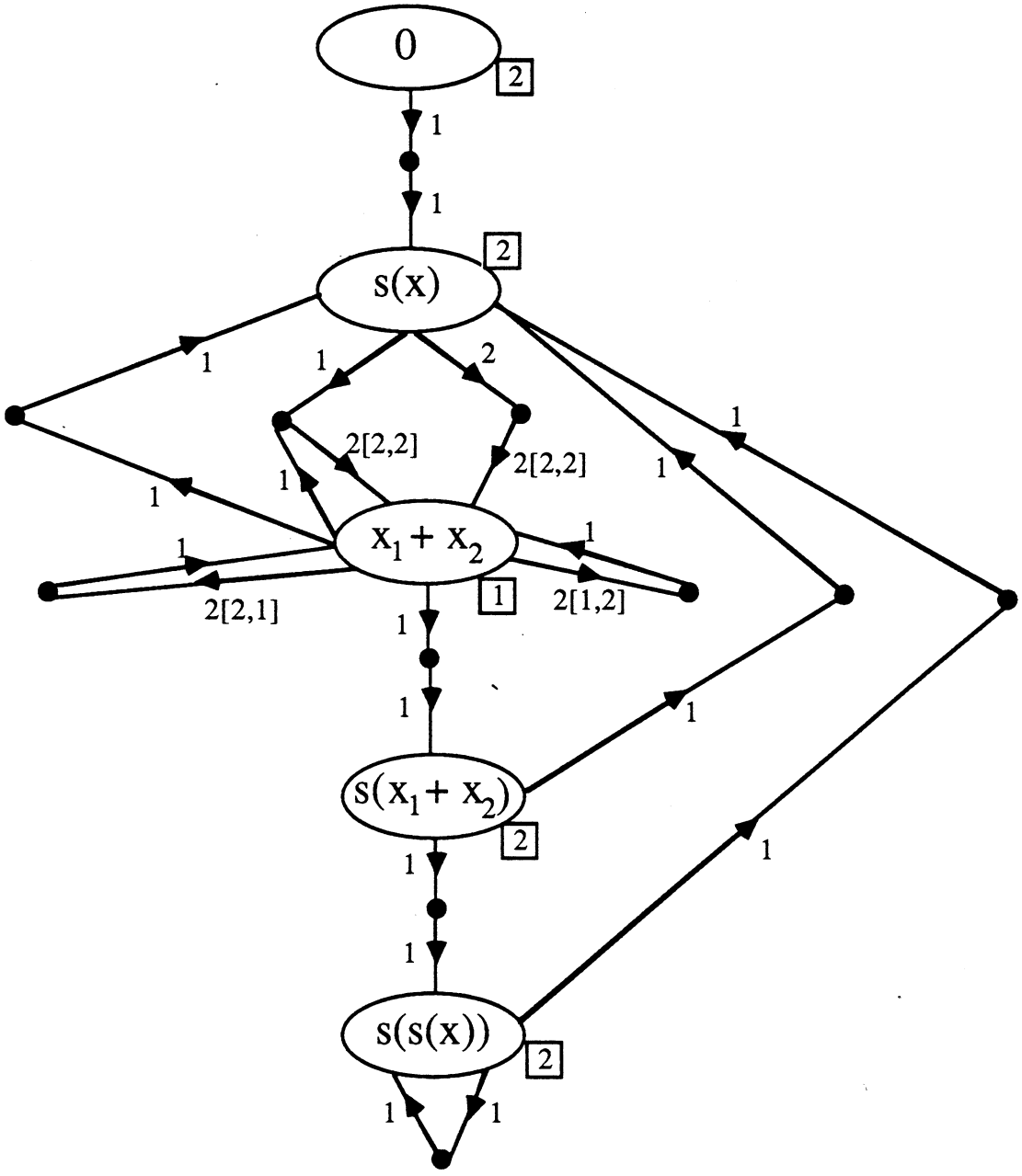

Figure 5.3: Transitions et nettoyage de grammaire

C'est cette méthode qui vient d'être décrite informellement sur l'exemple que nous allons employer dans le cas général.

5.3.2 Arbre de couverture

Les états C peuvent aussi être représentés par la liste des images des éléments de CT. Dans l'exemple 5.10,

$$C_0 = < \emptyset,\emptyset,\emptyset,\emptyset,\emptyset,\emptyset > \rightarrow_{\mathcal{G}} < \{0\},\emptyset,\emptyset,\emptyset,\emptyset,\emptyset > \rightarrow_{\mathcal{G}} < \{0\},\{s(0)\},\emptyset,\emptyset,\emptyset,\emptyset >$$

L'*arbre de couverture* de $\mathcal{G}$ est alors l'arbre (ou le graphe acyclique), éventuellement infini étiqueté par des états du calcul de $\mathcal{G}$ dont la racine est étiqueté par C_0 et tel que tous les fils s'un noeud étiqueté par C soient les états C' tels que $C \rightarrow_{\mathcal{G}} C'$. Chacun des arcs de cet arbre (ou graphe) est alors associé à une règle de $\mathcal{G}$ (la règle utilisée pour la réduction). Plus formellement, on peut dire que les positions de l'arbre de couverture sont des séquences de règles de grammaire.

Tout terme de $C(t,d)$ est aussi dans $[\![t,d]\!]$, pour chaque terme $u \in C(t,d)$ il existe donc une unique substitution σ de domaine $Var(t)$ telle que $t\sigma \equiv u$. On confondra donc parfois le terme $u \in C(t,d)$ et la substitution qui lui est associée.

5.3.3 Conditions suffisantes de réduction

Nous allons donner ici des conditions suffisantes sur un état C du calcul de $\mathcal{G}$ pour que C soit réductible par une règle R. L'idée étant que, s'il y a "suffisamment" de termes dans $C(t,d)$, les conditions de R doivent être remplies pour au moins un n-uple de termes de D_C. Autrement dit, nous cherchons ici à donner, en général, un moyen de calculer les chiffres inscrits dans les rectangles sur la figure 5.3.

Lemme 5.23 *Soit $R : NF_{t,d} \rightarrow u$ si δ un règle de $\mathcal{G}$ où $u \equiv \hat{t}[NF_{t_1,d_1},\ldots,NF_{t_n,d_n}]$. Soit C un état du calcul de $\mathcal{G}$.*

On suppose que $\exists\theta \in \Sigma_C$, $(u,\theta) \rightarrow_R (NF_{t,d},\sigma)$ et qu'il existe un indice i, une partition $Y \cup Z$ de $Var(t_i)$ et m_i substitutions $\sigma_1,\ldots,\sigma_{m_i}$ dans $C(t_i,d_i)$ telles que:

- $\forall j$, $\forall y \in Y$, $y\sigma_j \equiv y\theta$
- $\forall z \in Z$, $\forall i_1 \neq i_2$, $z\sigma_{i_1} \not\equiv z\sigma_{i_2}$,

Alors l'ensemble

$$\{j \leq m_i \mid \exists\phi \in \Sigma_g,\ (u,\sigma_j\theta) \rightarrow_R (NF_{t,d},\phi)\}$$

est de cardinal supérieur à $m_i - |\delta|$.

Preuve

Rappelons tout d'abord qu'on suppose (sans perdre de généralité) que les ensembles $Var(t_i)$ sont disjoints.

Soit $U = \{\sigma_j\theta \mid 1 \leq j \leq m_i\}$. Soit $z \neq v \in \delta$. Montrons qu'au plus une substitution $\psi \in U$ vérifie $z\psi \equiv v\psi$:

- Si $Var(t_i) \cap Var(z,v) = \emptyset$, alors, pour tout j, $z\sigma_j\theta \equiv z\theta$ et $v\sigma_j\theta \equiv v\theta$. Or l'hypothèse $(u,\theta) \rightarrow_R (NF_{t,d},\sigma)$ entraine que $z\theta \not\equiv v\theta$. Par conséquent, dans ce cas, aucune substitution de U ne vérifie $z\psi \equiv v\psi$.

- Si $z \in Var(t_i)$, alors, d'après la propriété 6 des problèmes simples, $Var(t_i) \cap Var(v) = \emptyset$ et donc $\forall \psi \in U$, $v\psi \equiv v\theta$. Deux cas se présentent alors:
  - $z \in Y$ et, par définition, $z\sigma_j\theta \equiv z\theta$ et donc $\forall \psi \in U$, $z\psi \not\equiv v\psi$.
  - $z \in Z$. Si i_0 vérifie alors $z\sigma_{i_0} \equiv v\theta$, par définition de Z, pour tout indice $i \neq i_0$, $z\sigma_i \not\equiv v\theta$. Par conséquent au plus une substitution $\psi \in U$ satisfait $z\psi \equiv v\psi$
- Si $Var(v) \cap Var(t_i) \neq \emptyset$. Alors, toujours par propriété des problèmes simples, $z \notin Var(t_i)$ et donc $\forall \psi \in U$, $z\psi \equiv z\theta$. Plusieurs cas se présentent alors:
  - $Z \cap Var(v) = \emptyset$. c'est-à-dire $Var(v) \subseteq Y$. Alors, à nouveau par définition de Y, $\forall \psi \in U$, $v\psi \equiv v\theta$ et donc $v\psi \not\equiv z\psi$.
  - $y \in Z \cap Var(v)$. Soit p une position de y dans v. Au plus une substitution σ_{i_0} vérifie $y\sigma_{i_0} \equiv v\theta/p$. D'où à nouveau le résultat souhaité

Il en résulte que au plus $|\delta|$ substitutions de U ne sont pas solutions de δ. Comme U est de cardinal m_i, nous obtenons le résultat souhaité. □

5.3.4 Substitutions dépendantes

La définition de dépendance (qui suit) est liée au résultat du lemme précédent: on s'intéresse aux substitutions dépendantes car, si elles sont en nombre suffisant, on peut prouver que certaines réductions sont possibles.

Définition 5.24 *m substitutions $\sigma_1, \ldots, \sigma_m \in C(t, d)$ sont dites* dépendantes *dans $C(t, d)$ si, pour toute partition de $Var(t)$ en deux sous-ensembles X_1 et X_2, X_1 étant non vide,*

- *ou bien, pour tout indice i_1, il existe un indice i_2 tel que*

$$\sigma_{i_1}|_{X_2} \neq \sigma_{i_2}|_{X_2}$$

- *ou bien,*

$$\exists z \in X_1,\ \exists i_1, i_2,\ i_1 \neq i_2,\ z\sigma_{i_1} \equiv z\sigma_{i_2}$$

Par convention, si X_2 est vide, les restrictions des substitutions à X_2 coincident. On peut aussi exprimer cette relation de dépendance de la façon suivante:

$\sigma_1, \ldots, \sigma_m$ sont *indépendantes* s'il existe une partition de $Var(t)$ en deux sous-ensembles X_1 et X_2 tels que X_1 est non vide et :

- pour toute variable z de X_2, les substitutions σ_i ont même valeur en z
- pour toute variable z de X_1, les termes $z\sigma_i$ sont tous distincts.

L'idée est alors de ne conserver dans $\mathcal{F}'(C)$ que des solutions dépendantes.

5.3.5 Calculs restreints

Dans la suite, $>_c$ désignera un ordre total sur les termes de $T(F)$ qui satisfait

$$C[u] >_c C[v] \Leftrightarrow u >_c v$$

pour tout terme $C \in T(F)$. De tels ordres peuvent être aisément construits à partir d'un ordre total sur les symboles de F.

Si $(t,d) \in CT$, on note $h_{t,d}$ la fonction qui associe à $u \in [\![t,d]\!]$ la longueur minimale d'une réduction $u \rightarrow NF_{t,d}$.

Si $(t,d) \in CT$, $>_{t,d}$ est l'ordre (total) défini sur $[\![t,d]\!]$ par:

$$\begin{array}{rcl} u >_{t,d} v & \Leftrightarrow & h_{t,d}(u) > h_{t,d}(v) \\ & & \text{ou bien } \; h(u) = h_{t,d}(v) \text{ et } u >_c v \end{array}$$

Cet ordre peut aussi être vu comme un ordre sur les substitutions en confondant (comme nous le faisons) les termes de $[\![t,d]\!]$ et les substitutions fermées de domaine $Var(t)$ correspondantes.

Nous pouvons maintenant définir un *calcul restreint* de $\mathcal{G}$. Soit λ une application de CT dans l'ensemble des entiers naturels non nuls, $\mathcal{F}_\lambda$ est l'application (appelée calcul restreint de $\mathcal{G}$) de l'ensemble des états dans lui-même telle que, pour tout état C et tout $(t,d) \in CT$:

- si $|\mathcal{F}(C)(t,d)| \leq \lambda(t,d)$, $\mathcal{F}_\lambda(C)(t,d) = \mathcal{F}(C)(t,d))$.
- Sinon, soit $\mathcal{F}(C)(t,d) = \{\sigma_1, \ldots, \sigma_n\}$ avec $\sigma_1 <_{t,d} \sigma_2 <_{t,d} \ldots <_{t,d} \sigma_n$. Alors, $\sigma_i \in \mathcal{F}_\lambda(C)(t,d)$ ssi $i \leq \lambda(t,d)$ ou bien $i > \lambda(t,d)$ et $1 + \lambda(t,d)$ substitutions quelconques dans $(\{\sigma_1, \ldots, \sigma_{i-1}\} \cap \mathcal{F}_\lambda(C)(t,d)) \cup \{\sigma_i\}$ sont dépendantes.

Il est ainsi possible de construire $\mathcal{F}_\lambda(C)(t,d)$ en énumérant les substitutions de $\mathcal{F}(C)(t,d)$ par ordre croissant et en testant pour chacune d'elles la dépendance avec les substitutions déjà calculées. Remarquons aussi que $1+\lambda(t,d)$ substitutions quelconques dans $\mathcal{F}_\lambda(C)(t,d)$ sont dépendantes.

Lemme 5.25 *Si* $C_0 \rightarrow_{\mathcal{G}}^* C \rightarrow_R C'$ *alors, pour tout non-terminal* (t,d),

$$\mathcal{F}_\lambda(C)(t,d) \subseteq \mathcal{F}_\lambda(C')(t,d)$$

Preuve

Cela résulte du fait que, si $u \in C'(t,d) - C(t,d)$ et $v \in C(t,d)$, alors $u >_{t,d} v$ puisque $h_{t,d}(u) > h_{t,d}(v)$. □

C'est essentiellement pour obtenir cette propriété que nous avons introduit l'ordre $>_{t,d}$.

5.3.6 $\mathcal{F}_\lambda(C)(t,d)$ est fini

Les deux lemmes qui suivent ont alors pour objectif de prouver que $\mathcal{F}_\lambda(C)(t,d)$ est fini pour tous C, λ, t, d. Cette propriété est fondamentale pour la terminaison de l'algorithme.

Lemme 5.26 *Soit $\mathcal{R}$ une relation m-aire de graphe $G \subseteq A_1 \times \ldots \times A_m$. On suppose que G possède la propriété suivante:*

Si G_1 est une partie de G et E un sous-ensemble de $\{1,\ldots,m\}$ tels que:

- $\forall i \in E, \forall a,b \in G_1,\ a_i = b_i$
- $\forall i \notin E, \forall a,b \in G_1,\ a_i \neq b_i$

alors G_1 a moins de N éléments.

Alors G a moins de N^{2^m-1} éléments.[14]

Preuve

On prouve le lemme par récurrence sur m. Si $m = 1$, le résultat est trivial. S'il est vrai pour $m-1$, soit $A_1^1 = \{a_1 \in A_1 \mid \exists (a_2,\ldots,a_n),(a_1,\ldots,a_n) \in G\}$. Notons alors f une application de A_1^1 dans $A_2 \times \ldots \times A_n$ qui associe à chaque élément a_1 un tuple $(a_2,\ldots,a_n)$ tel que $(a_1,\ldots,a_n) \in G$. On note $H = f(A_1^1)$. H vérifie les mêmes hypothèses que G. En effet, si E' est une partie de $\{2,\ldots,m\}$ et H_1 un sous-graphe de H tels que

- $\forall a,b \in H_1, \forall i \in E',\ a_i = b_i$
- $\forall a,b \in H_1, \forall i \notin E',\ a_i \neq b_i$

alors, à tout $a \in H_1$ on associe un élément de G de la forme (a_1, a). (a_1 est un antécédent de a par f). Soit G_1 la partie de G ainsi obtenue. Soit encore $E = E'$. Comme, pour tous $a,b \in G_1$, $a_1 \neq b_1$ par construction, G_1 et E vérifient les hypothèses du lemme et possède donc moins de N éléments. Par conséquent, il en est de même de H_1.

Par hypothèse de récurrence, H a donc moins de $N^{2^{m-1}-1}$ éléments.

Si l'on choisit maintenant $E = \{2,\ldots,m\}$, on obtient que, pour tout $x \in H$, $|f^{-1}(x)| \leq N$. Il en résulte que $|A_1^1| \leq N^{2^{m-1}-1} * N$.

Pour tout $a_1 \in A_1^1$ notons maintenant G_{a_1} le graphe contenu dans $A_2 \times \ldots \times A_n$ défini par:

$$(a_2,\ldots,a_n) \in G_{a_1} \Leftrightarrow (a_1,\ldots,a_n) \in G$$

[14] Nous conjecturons que cette borne de N^{2^m-1} peut être ramenée à N^m (et est donc loin d'être optimale). Dans le cas où $m = 2$, on peut prouver que G est effectivement de cardinal inférieur à N^2: c'est une conséquence du théorème de Kõnig sur les graphes simples (cf [Ber83] par exemple).

En effet, si l'on choisit tout d'abord $E = \emptyset$, les hypothèses du lemme signifient exactement que tout couplage de G est de cardinal inférieur à N. D'après le théorème de Kõnig il existe alors un transversal de cardinal inférieur à N. D'autre part, si l'on choisit successivement $E = \{1\}$ et $E = \{2\}$, on obtient que le degré maximal d'un noeud du graphe est N. Le cardinal de G étant inférieur à la somme des degrés des noeuds d'un transversal minimum, on obtient le résultat.

Il n'y a pas, à notre connaissance, de généralisation de ce théorème aux hypergraphes. De toutes façons, il est possible que l'extension aux hypergraphes du théorème de Kõnig soit fausse sans pour autant infirmer notre conjecture.

G_{a_1} possède les mêmes propriétés que G (à $E' \subseteq \{2,\ldots,n\}$ on fait cette fois correspondre $E = E' \cup \{1\}$). Et donc, par hypothèse de récurrence, $|G_{a_1}| \leq N^{2^{m-1}-1}$. Comme $G = \bigcup_{a_1 \in A_1^1} G_{a_1}$, on déduit les inégalités:

$$|G| \leq \sum_{a_1 \in A_1^1} |G_{a_1}| \leq |A_1^1| * N^{2^{m-1}-1} \leq N * N^{2^{m-1}-1} * N^{2^{m-1}-1} = N^{2^m-1}$$

□

Lemme 5.27 *Pour toute application λ et tout $N = NF_{t,d} \in CT$, il existe un nombre $K_\lambda(N)$ $(= \lambda(t,\delta)^{2^m-1}$ où m est le nombres de variables de t) tel que, pour tout état C,*

$$|\mathcal{F}_\lambda(C)(t,\delta)| \leq K_\lambda(N)$$

Preuve

Par définition, M substitutions de $\mathcal{F}_\lambda(C)(t,\delta)$ sont dépendantes dès que $M > \lambda(t,\delta)$. Si m est le nombre de variables de t: $Var(t) = \{x_1,\ldots,x_m\}$, soit R la relation m-aire dont le graphe G est l'ensemble $\{(x_1\sigma,\ldots,x_m\sigma) \in T(F)^m \mid \sigma \in \mathcal{F}_\lambda(C)(t,\delta)\}$.

Soit alors G_1 une partie de G et X_1, X_2 une partition de $Var(t)$ en deux sous-ensembles tels que X_1 est non-vide. Chaque élément de G_1 définit une unique substitution de $\mathcal{F}_\lambda(C)(t,\delta)$. Si G_1 est de cardinal supérieur à $\lambda(t,\delta)$, la propriété de dépendence des substitutions de $\mathcal{F}_\lambda(C)(t,\delta)$ entraine que:

- ou bien $\exists z \in X_2, \exists \theta_1,\theta_2 \in G_1, z\theta_1 \not\equiv z\theta_2$.
- ou bien $\exists z \in X_1, \exists \theta_1,\theta_2 \in G_1, \theta_1 \neq \theta_2, z\theta_1 \equiv z\theta_2$

Les hypothèses du lemme 5.26 sont alors satisfaites. Il en résulte que G (qui a même nombre d'éléments que $\mathcal{F}_\lambda(C)(t,\delta)$) est de cardinal inférieur à $(\lambda(t,\delta))^{2^m-1}$. □

5.3.7 Réductions et calculs restreints

Les résultats qui suivent ont pour but de prouver que, tout "nouvel" élément dans $\mathcal{F}_\lambda(C)(t,\delta)$ provient nécessairement d'un "nouvel" élément dans $\mathcal{F}_\mu(C)(u,d)$. Autrement dit, il n'est pas nécessaire de considérer d'autres termes que ceux de $\mathcal{F}_\lambda(C)$ pour un λ bien choisi.

Lemme 5.28 *Soient C_1, C_2, C_3 trois noeuds consécutifs de l'arbre de couverture: $C_1 \rightarrow_{R_1} C_2 \rightarrow_{R_2} C_3$. On suppose qu'il existe un ensemble $A = \{\psi_1,\ldots,\psi_k\} \subseteq C_2(t,\delta)$ tel que*

$$k \geq \lambda(t,\delta) + \frac{|\mathcal{F}_\lambda(C_1)(t,\delta)|}{\lambda(t,\delta)} - 1$$

On suppose de plus qu'il existe une variable $x \in Var(t)$ et une substitution $\sigma \in C_3(t,\delta) - C_2(t,\delta)$ telle que

$$\forall i,\ \psi_i|_{Var(t)-\{x\}} = \sigma|_{Var(t)-\{x\}}$$

Alors, $\sigma \notin \mathcal{F}_\lambda(C_2)(t,\delta)$.

Preuve

Notons tout d'abord que, s'il existe $\lambda(t,\delta)$ substitutions $\theta_1,\ldots,\theta_M$ dans $A \cap \mathcal{F}_\lambda(C_1)(t,\delta)$ ($\subseteq \mathcal{F}_\lambda(C_2)(t,\delta)$), alors on a le résultat souhaité. En effet, $\theta_1,\ldots,\theta_M,\sigma$ ne sont pas dépendantes car il suffit de choisir $X_1 = \{x\}$ et $X_2 = Var(t) - \{x\}$ pour obtenir les deux conditions :

- les substitutions $\theta_1,\ldots,\theta_M,\sigma$ coincident sur les variables de X_2
- Elles prennent des valeurs toutes distinctes en x

Considérons désormais l'ensemble $A' = A - \mathcal{F}_\lambda(C_1)(t,\delta)$. A' est de cardinal supérieur à $|\mathcal{F}_\lambda(C_1)(t,\delta)|/\lambda(t,\delta)$. Si $\phi \in A'$, par définition de la non dépendance, il existe $\lambda(t,\delta)$ substitutions $\theta_1,\ldots,\theta_M \in \mathcal{F}_\lambda(C_1)(t,\delta)$ et une partition X_1, X_2 de $Var(t)$ telles que :

- les substitutions θ_i coincident avec ϕ sur X_2
- pour tout $z \in X_1$, pour tous indices distincts i et j, $z\theta_i \not\equiv z\phi$ et $z\theta_i \not\equiv z\theta_j$

Ainsi, à chaque élément $\phi \in A'$ on peut associer

- Le sous-ensemble X_ϕ de $Var(t)$ des variables sur lesquelles les substitutions θ_i coincident avec ϕ
- l'ensemble E_ϕ des $\lambda(t,\delta)$ substitutions $\theta_1,\ldots\theta_M$.

Remarquons d'autre par que, comme $\sigma \notin C_2(t,\delta)$, $x\sigma \not\equiv x\theta$ pour toute substitution $\theta \in C_2(t,\delta)$. En effet, pour $y \not\equiv x$, $y\theta \equiv y\sigma$ pour au moins une substitution de $C_2(t,\delta)$.[15]

Raisonnons maintenant par l'absurde et supposons que $\sigma \in \mathcal{F}_\lambda(C_2)(t,\delta)$. Alors, pour toute substitution $\phi \in A'$,

- ou bien $\exists\theta \in E_\phi,\ \sigma|_{X_\phi} \neq \theta|_{X_\phi}$
- ou bien $\exists\theta \in E_\phi,\ \exists z \in Var(t) - X_\phi,\ z\theta \equiv z\sigma$

Mais, par définition de X_ϕ, pour toute variable $z \in Var(t) - X_\phi$, $z\phi \not\equiv z\theta$. Par propriété de ϕ, on a de plus, pour $z \not\equiv x$, $z\phi \equiv z\sigma$. Enfin, comme vu ci-dessus, $x\theta \not\equiv x\sigma$ et donc, $\forall z \in Var(t) - X_\phi$, $z\theta \not\equiv z\sigma$. σ ne peut donc vérifier la deuxième propriété ci-dessus: σ vérifie nécessairement la première des deux propriétés.

Mais, comme $\sigma|_{Var(t)-\{x\}} = \phi|_{Var(t)-\{x\}}$, la propriété $\sigma|_{X_\phi} \neq \theta|_{X_\phi}$ entraine que $x \in X_\phi$. On en déduit que, pour tout $\phi \in A'$ et tout $\theta \in E_\phi$, $x\theta \equiv x\phi$. Les ensembles E_ϕ sont donc tous disjoints puisque deux substitutions distinctes de A' prennent des valeurs distinctes sur x. Il en résulte que $\mathcal{F}_\lambda(C)(t,\delta)$ est de cardinal supérieur à

$$\sum_{\phi \in A'} |E_\phi| \ \geq\ \lambda(t,\delta) * |\mathcal{F}_\lambda(C_1)(t,\delta)| \,/\, \lambda(t,\delta)$$

Comme σ n'est pas dans cet ensemble, il en résulte que

$$|\mathcal{F}_\lambda(C_1)(t,\delta)| \ >\ |\mathcal{F}_\lambda(C_1)(t,\delta)|$$

[15] Rappelons que, si $u_1,\ldots,u_n \in T(F)$ vérifient $\forall x_i \in \{x_1,\ldots,x_n\} = Var(t), \exists\theta \in \mathcal{F}(C)(t,\delta), x_i\theta \equiv u_i$ et si $\{x_1 \rightarrow u_1,\ldots,x_n \rightarrow u_n\} \in [\![t,\delta]\!]$, alors $\{x_1 \rightarrow u_1,\ldots,x_n \rightarrow u_n\} \in \mathcal{F}(C)(t,\delta)$.

Ce qui est absurde. □

Pour simplifier, nous supposerons désormais que, pour tout t, $\widehat{t} \equiv t$. Cette supposition est licite puisque $Pos(t) - \{\epsilon\}$ est un ensemble complet de positions de t.

Le lemme suivant établit le résultat annoncé plus haut: une "nouvelle" substitution dans $\mathcal{F}_\lambda$ provient d'une substituion dans $\mathcal{F}_\mu$. Malheureusement, μ est beaucoup plus grand que λ et nous n'aurons donc pas terminé. Il faut noter qu'il existe vraisemblablement des améliorations de la borne de clacul de μ en fonction de λ. Certaines améliorations substantielles sont proposées dans des cas particuliers dans la section 5.4.

Lemme 5.29 *Soient* C_1, C_2, C_3 *trois noeuds consécutifs de l'arbre de couverture. Si bien que* $C_1 \rightarrow_{R_1} C_2 \rightarrow_{R_2} C_3$ *avec* $R_2 : N \rightarrow w$ *Si* d. *On suppose que*

$$\sigma \in \mathcal{F}_\lambda(C_2)(t,\delta) \cap (C_3(t,\delta) - \mathcal{F}(C_1)(t,\delta))$$

Alors il existe un non-terminal $NF_{u,d'}$ *tel que:*

$$\exists \theta \in \Sigma_{C_1} \exists \phi \in \mathcal{F}_\mu(C_1)(u,d') - C_1(u,d'),\ (w, \phi\theta) \rightarrow_{R_2} (NF_{t,\delta}, \sigma)$$

où μ *est l'application coincidant avec* λ *sauf en* $NF_{u,d'}$ *où elle prend la valeur*[16]

$$\mu(u,d') \ = \ |d| + \lambda(t,\delta) + \frac{|\mathcal{F}_\lambda(C_2)(t,\delta)|}{\lambda(t,\delta)} - 1$$

Preuve

La condition $\mathcal{F}_\lambda(C_2)(t,\delta) \cap C_3(t,\delta) \neq \emptyset$ entraine que la règle R_2 est de la forme:

$$NF_{t,\delta} \rightarrow t[NF_{t_1,d_1}, \ldots, NF_{t_n,d_n}] \text{ Si } d$$

La condition $\mathcal{F}_\lambda(C_2)(t,\delta) - \mathcal{F}(C_1)(t,\delta) \neq \emptyset$ entraine d'autre part que la règle R_1 a effectivement modifié l'un des $C(t_i,d_i)$. (i.e. $\exists i, C_2(t_i,d_i) \neq C_1(t_i,d_i)$). Supposons par exemple (sans perdre de généralité) que cet indice i est 1 et soit $\phi_1 \in C_2(t_1,d_1) - C_1(t_1,d_1)$.

Raisonnons par l'absurde et supposons que $\phi_1 \notin \mathcal{F}_\mu(C_1)(t_1,d_1)$. Par définition, il existe alors m substitutions ($m \geq \mu(t_1,d_1)$) $\theta_1, \ldots, \theta_m \in \mathcal{F}_\mu(C_1)(t_1,d_1)$ telles que $\theta_1, \ldots, \theta_m, \theta_{m+1} = \phi_1$ ne soient pas dépendantes dans $\mathcal{F}(C_1)(t_1,d_1)$. Il existe alors (par définition de la non dépendance) un sous-ensemble X_0 de $Var(t_1)$ tel que:

- $\forall i,\ \theta_i|_{X_0} = \phi_1|_{X_0}$
- $\forall i_1 \neq i_2, \forall z \notin X_0,\ z\theta_{i_1} \not\equiv z\theta_{i_2}$

Les hypothèses du lemme 5.23 sont alors satisfaites: il existe un sous-ensemble A de $\{\theta_1, \ldots, \theta_M\}$ de cardinal supérieur à $\mu(u,d') - |d|$ tel que, pour tout $\theta_i \in A$, il existe une substitution psi_i telle que $(w, \theta_i\theta) \rightarrow_{R_2} (NF_{t,\delta}, \psi_i)$.

Par choix de l'indiçage, on obtient ainsi un ensemble $\{\psi_1, \ldots, \psi_k\} \subseteq C_3(t,\delta)$ de substitutions telles que $\psi_1 <_{t,d} \ldots <_{t,d} \psi_k$ avec $k \geq \mu(u,d') - |d|$. Notre objectif est alors

[16] Là encore, nous conjecturons que la valeur de μ peut être améliorée et qu'il suffit de choisir $\mu(u,d) = \lambda(t,\delta) + |d|$. De nouveau, cette borne est très inférieure à la borne proposée dans le lemme.

de montrer que les substitutions ψ_i sont dans $C_2(t,\delta)$ afin de pouvoir appliquer le lemme 5.28.

Nous pouvons tout d'abord remarquer que, comme $\theta_1,\ldots,\theta_M <_{t_1,d_1} \phi_1, \psi_1,\ldots,\psi_k <_{t,d} \sigma$ et que, pour tout i, $\psi_i|_{Var(t)-\{x_1\}} = \sigma|_{Var(t)-\{x_1\}}$.

Soit r le plus petit indice tel que $\psi_r \notin \mathcal{F}_\lambda(C_2)(t,\delta)$. Un tel indice existe bien sinon, comme $\psi_1,\ldots,\psi_k,\sigma$ sont indépendantes et en nombre supérieur à $1+\lambda(t,\delta)$, on aurait $\sigma \notin \mathcal{F}_\lambda(C_2)(t,\delta)$.

Soient donc $\theta'_1,\ldots\theta'_p,\theta'_{p+1} = \psi_i$ $(igeqr)$ telles que $\theta'_1,\ldots,\theta'_p \in \mathcal{F}_\lambda(C_2)(t,\delta)$ et $\theta'_1,\ldots,\theta'_{p+1}$ sont indépendantes; il existe un sous-ensemble Y de $Var(t)$ tel que:

- $\forall j,\ \theta'_j|_Y = \psi_i|_Y$
- $\forall j_1 \neq j_2,\ \forall z \notin Y,\ z\theta'_{j_1} \not\equiv z\theta'_{j_2}$

Deux cas se présentent:

Premier cas: $x_1 \notin Y$

Dans ce cas, pour tout j, $x\theta'_j \not\equiv x\sigma$. En effet;

- ou bien θ'_j est l'une des substitutions ψ_l et cela résulte de $\psi_l \neq \sigma$
- ou bien θ'_j n'est pas l'un des ψ_l et dans ce cas $\theta'_j \in C_2(t,\delta)$. Mais alors $x_1\sigma \equiv t_1\phi_1 \notin C_1(t_1,d_1)$ alors que $x_1\theta'_j \in C_1(t_1,d_1)$. Ce qui est absurde

Ce premier cas ne peut donc avoir lieu.

Deuxième cas : $x_1 \in Y$

Alors $x_1\theta'_1 \equiv \ldots \equiv x_1\theta'_p \equiv x_1\psi_i$. Comme, pour au moins un indice l, θ'_l n'est pas l'un des ψ_j, pour au moins un l, $\theta'_l \in C_2(t,\delta)$. Il en résulte que $\psi_1,\ldots,\psi_i \in C_2(t,\delta)$.

En choisissant $i = k$ on obtient ainsi $\psi_1,\ldots,\psi_k \in C_2(t,\delta)$. Il est alors possible d'appliquer le lemme 5.28: on obtient le résultat absurde que $\sigma \notin \mathcal{F}_\lambda(C_2)(t,\delta)$. Donc $\phi_1 \in \mathcal{F}_\mu(C_1)(t_1,d_1)$. □

Ce dernier résultat n'est pas complètement satisfaisant, car, contrairement à ce qu'on pourrait penser,il ne suffit pas pour conclure. En effet, si $\mu > \lambda$ et, par exemple, $(u,d') \equiv (t,\delta)$, il ne nous est pas possible de donner une borne au nombre d'éléments à calculer dans $\mathcal{F}'(C)(u,d)$.

Le résultat qui suit permet de montrer que, dans ce cas, "quelque chose" a décrû en passant de λ à μ. La figure 5.4 montre intuitivement le résultat lorsque le nombre de variables de u est égal à 2: la "non satisfaction" des diséquations limite les valeurs prises par les variables à la zone hachurée. Les autres valeurs sont "passées" et ont produit de nouveaux éléments dans $\mathcal{F}_\lambda(C)(t,\delta)$. Ce sont les points éparpillés hors de la zone hachurée. Par la suite, si l'on passe deux fois par le non-terminal (u,d), ou bien l'on enlève un point dans la zone non hachurée, sans changer celle-ci, ou bien on réduit la zone hachurée. Cela permet de borner le nombre de fois où l'on passe par le non terminal (u,d).

Lemme 5.30 *Soient C_1,C_2,C_3 trois noeuds consécutifs de l'arbre de couverture. (Si bien que $C_1 \rightarrow_{R_1} C_2 \rightarrow_{R_2} C_3$ où $R_2 : NF_{t,\delta} \rightarrow w$ si d). On suppose que $\sigma \in C_3(t,\delta) -$*

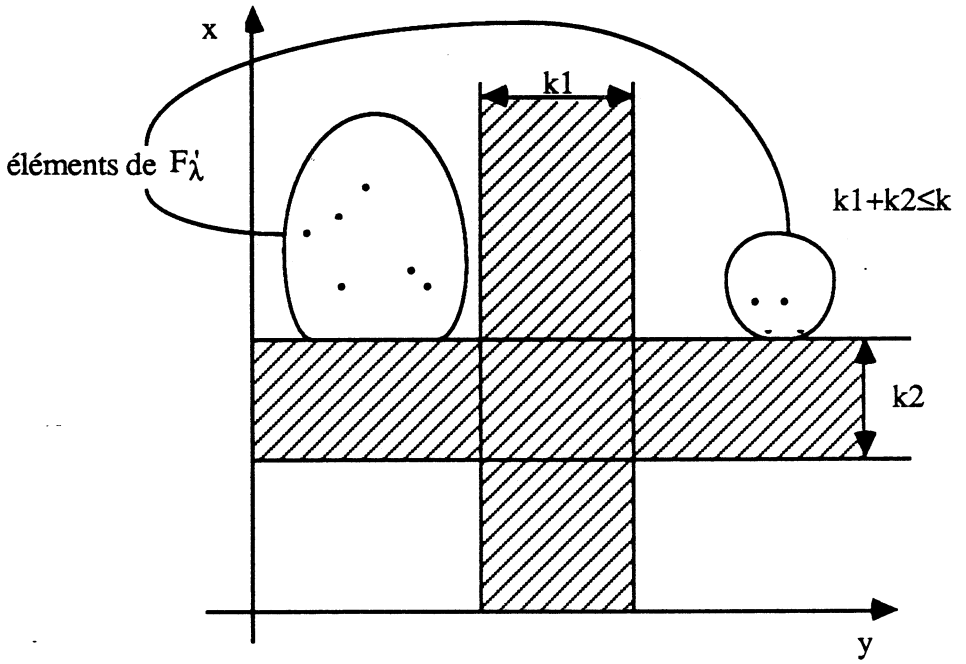

Figure 5.4: Illustration du lemme 5.30

$\mathcal{F}(C_1)(t,\delta)$ et $\sigma \in \mathcal{F}_\lambda(C_2)(t,\delta)$. Alors, il existe $NF_{u,d'} \in CT$, il existe des entiers strictement positifs $k_1,\ldots,k_m$ (avec $m = |Var(u)|$) tels que $k_1 + \ldots + k_m \leq |d|$, il existe un sous-ensemble E de $C_2(u,d')$, de cardinal strictement inférieur à

$$\lambda(t,\delta) + \frac{|\mathcal{F}_\lambda(C_1)(t,\delta)|}{\lambda(t,\delta)} - 1$$

il existe des sous-ensembles $E_1,\ldots,E_m$ de $T(F)$ tels que:

- $\forall i,\ |E_i| \leq k_i$
- $\forall \theta \in C_2(u,d')$, *ou bien $\theta \in E$ ou bien il existe une variable y de u et un indice i tels que $y\theta \in E_i$.*

Le résultat de ce lemme est illustré par la figure 5.4.

Preuve

Comme précédemment, R_2 peut s'écrire:

$$NF_{t,\delta} \rightarrow t[NF_{t_1,d_1},\ldots,NF_{t_n,d_n}] \text{ si } d$$

Comme précédemment, on peut aussi supposer que NF_{t_1,d_1} est le seul non terminal tel que $C_2(t_1,d_1) \neq C_2(t_1,d_1)$. Soit ϕ_1 telle que $(w,\phi_1\theta) \rightarrow_{R_2} (NF_{t,\delta},\sigma)$, $Dom(\phi_1) = Var(t_1)$, $\phi_1 \in C_2(t_1,d_1) - C_1(t_1,d_1)$ et $Dom(\theta) = Var(t_2,\ldots,t_n)$.

Soit D une des formes irréductibles de $d\theta$. D est une conjonction de diséquations dont les membres gauches sont des variables de t_1 et les membres droits sont des termes clos.

On définit alors E_i comme l'ensemble des membres droits dont les membres gauches sont $y_i \in Var(t_1)$. Comme $|D| \leq |d|$, on obtient la condition $k_1 + \ldots + k_m \leq |d|$. Soit

$$E = \{\phi \in C_2(t_1, d_1) \mid \forall y_i \in Var(t_1), y_i\phi \notin E_i\}$$

On obtient alors la condition $\forall \theta \in C_2(t_1, d_1)$, $\theta \in E$ ou bien $\exists y_i \in Var(t_1)$, $y_i\theta \in E_i$.

Pour toute substitution $\phi \in E$, $\phi\theta$ est, par construction, une solution de d. On peut donc asocier à chaque substitution de E une substitution de $C_3(t, \delta)$ qui coincide avec σ sur toutes les variables de t autres que x_1. D'après le lemme 5.28 et puisque $\sigma \in \mathcal{F}_\lambda(C_2)(t, \delta)$, E est alors de cardinal strictement inférieur à $\frac{|\mathcal{F}_\lambda(C_1)(t,\delta)|}{\lambda(t,\delta)} + \lambda(t, \delta) - 1$. □

Le lemme suivant établit formellement la propriété de décroissance annoncée.

Lemme 5.31 *Soient C_1, C_2, C_3, C_4 quatre noeuds de l'arbre de couverture tels que:*

- $C_1 \rightarrow_{R_1} C_2 \rightarrow_{\mathcal{G}}^{+} C_3 \rightarrow_{R_2} C_4$
- $\sigma \in \mathcal{F}_{\lambda'}(C_2)(t, \delta) - \mathcal{F}(C_1)(t, \delta)$
- $\theta \in \mathcal{F}_\lambda(C_4)(t, \delta) - \mathcal{F}(C_3)(t, \delta)$
- $C_2(u, d') \neq C_1(u, d')$ *et* $C_3(u, d') \neq C_4(u, d')$
- $$\exists k_1, \ldots, k_m \in \mathbf{N}, \exists E_1, \ldots, E_m \subseteq T(F), \forall i, |E_i| \leq k_i, \\ \exists E \subseteq C_4(u, d'), |E| \leq \beta, \forall \phi \in C_4(u, d'), \\ \phi \in E \text{ ou } \exists y_i \in Var(u), y_i\phi \in E_i$$

Alors

$$\exists k'_1, \ldots, k'_m \in \mathbf{N}, \exists E'_1, \ldots, E'_m \subseteq T(F), \forall i, |E'_i| \leq k'_i, \exists E' \subseteq C_2(u, d'), |E'| \leq \beta'$$

tels que:

- $\forall \phi \in C_2(u, d'), \phi \in E'$ *ou* $\exists y_i \in Var(u), y_i\phi \in E'_i$
- – *ou bien* $k'_1 + \ldots + k'_m < k_1 + \ldots + k_m$ *et* $\beta' \leq \lambda(t, \delta)^{2^{|Var(t)|-1}-2} + \lambda(t, \delta) + \beta$
  – *ou bien* $k'_1 + \ldots + k'_m \leq k_1 + \ldots + k_m$ *et* $\beta' < \beta$

Preuve

Soient $\phi_1, \theta_1, \phi_2, \theta_2$ tels que:

- $(w, \phi_1\theta_1) \rightarrow_{\mathcal{G}} (NF_{t,\delta}, \sigma)$
- $(w', \phi_2\theta_2) \rightarrow_{\mathcal{G}} (NF_{t,\delta}, \theta)$
- $\phi_1 \in C_2(u, d') - C_1(u, d')$
- $\phi_2 \in C_4(u, d') - C_3(u, d')$

Deux cas se présentent alors:

Premier cas : $E \not\subseteq C_2(u,d')$

Il suffit alors de choisir $E' = E \cap C_2(u,d')$ et $E'_j = E_j$ pour tout j. On a alors $|E'| < |E|$ et $k_1 + \ldots + k_m = k'_1 + \ldots + k'_m$ et $\forall \phi \in C_2(u,d'), \exists i, y_i\phi \in E_i$ ou $\phi \in E$ par hypothèse. Donc, $\forall \phi \in C_2(u,d'), \exists i, y_i\phi \in E'_i$ ou $\phi \in E'$.

Deuxième cas: $E \subseteq C_2(u,d')$

Alors $\phi_2 \notin E$ et donc $\exists j,\ y_j\phi_2 \in E_j$. On pose alors $E'_j = E_j - \{y_j\phi_2\}$ et $E'_i = E_i$ sinon. Dans ce cas $k'_1 + \ldots + k'_m < k_1 + \ldots + k_m$ et, $\forall \phi \in C_2(u,d')$, comme $\phi \in C_4(u,d')$:

- ou bien $\exists i,\ y_i\phi \in E'_i$
- ou bien $\phi \in E$
- ou bien $y_j\phi \equiv y_j\phi_2$

Mais ces dernières substitutions sont au plus en nombre $\lambda(t,\delta) + \frac{|\mathcal{F}_\lambda(C_3)(t,\delta)|}{\lambda(t,\delta)} - 1$ d'aorès le lemme 5.28. On obtient alors le résultat à l'aide de la majoration du lemme 5.27.

□

5.3.8 Définition de $\mathcal{F}'$

$\mathcal{F}'$ n'a pas encore été défini. Nous allons le définir comme un $\mathcal{F}_\lambda$ pour λ bien choisi. Pour cela, nous avons encore besoin d'un dernier résultat technique:

Lemme 5.32 *Soit* $CT = \{(t_1,d_1),\ldots,(t_n,d_n)\}$. *Si* $\mathcal{F}_1(\mathcal{F}^r(C_0))(t_i,d_i) \not\subseteq \mathcal{F}^r(C_0)(t_i,d_i)$ *alors* $r \geq \nu(K)$ *où:*

- *pour tout* i, k_i *est le nombre maximal de diséquations dans une règle dont le membre gauche est* NF_{t_i,d_i}.
- $K = \sum_{1 \leq i \leq n} k_i$
- ν *est défini par récurrence par:* $\nu(0) = 1$ *et* $\nu(i+1) = \rho_{n*\nu(i)}$
- ρ_i *est défini par récurrence par:* $\rho_0 = 0$ *et* $\rho_{i+1} = \rho_i^{2^{m-1}-2} + \rho_i - 1$ *où* m *est le nombre maximal de variables d'un non-terminal.*

Preuve

Soit C un état, $NF_{t_i,d_i} \in NT$ et $\{y_1,\ldots,y_m\} = Var(t_i)$. A tous ensembles $E_1,\ldots,E_m \subseteq T(F)$ et $E \subseteq C(t_i,d_i)$ tels que, pour toute substitution $\sigma \in C(t_i,d_i)$, ou bien $\sigma \in E$, ou bien $\exists j, y_j\sigma \in E_j$ on associe le couple $(|E_1| + \ldots + |E_m|, |E|)$. La fonction ainsi définie admet un minimum sur $\mathbf{N}^2$ (muni de l'ordre lexicographique), qu'elle atteint pour $H_C(t_i,d_i) = (E_{1,i},\ldots,E_{m,i},E_i)$. On note aussi $M_C(t_i,d_i) = (k_{i,C},h_{i,C})$ ce minimum. A chaque état C on peut ainsi associer les ensembles $H(C) = (H_C(t_1,d_1),\ldots,H_C(t_n,d_n))$ et $M(C) = (M_C(t_1,d_1),\ldots,M_C(t_n,d_n))$.

Soit alors, pour tout i, $C_i = \mathcal{F}^i(C_0)$. Supposons que $\mathcal{F}_1(C_r)(t_i,d_i)$ n'est pas contenu dans $C_r(t_i,d_i)$. On définit alors par récurrence la suite de non-terminaux N_j et la suite d'entiers μ_j de sorte que, pour tout j, $\mathcal{F}_{\mu_j}(C_{r-j})(N_j)$ n'est pas contenu dans $C_{r-j}(N_j)$:

- $NT_0 = (t_i, d_i)$ et $\mu_0 = 1$
- D'après le lemme 5.29, si $\mathcal{F}_{\mu_j}(C_{r-j})(N_j)$ n'est pas contenu dans $C_{r-j}(N_j)$, il existe un entier μ_{j+1} et un non-terminal N_{j+1} qui vérifient

$$\mathcal{F}_{\mu_{j+1}}(C_{r-j-1})(N_{j+1}) \not\subseteq C_{r-j-1}(N_{j+1})$$

  On peut aussi choisir $\mu_{j+1} = \mu_j^{2^{m-1}-2} + \mu_j - 1$ où m est le nombre de variables de N_j, d'après les lemmes 5.27 et 5.29.

De plus, d'après le lemme 5.30, si $N_j = NF_{t_i,d_i}$, alors $k_{i,C_j} \leq k_i$ et $h_{i,C_j} \leq \mu_j$. Et, d'après le lemme 5.31, si $N_{j_1} = N_{j_2} = NF_{t_i,d_i}$ avec $j_1 < j_2$, alors

- ou bien $k_{i,C_{j_1}} < k_{i,C_{j_2}}$
- ou bien $k_{i,C_{j_1}} = k_{i,C_{j_2}}$ et $h_{i,C_{j_1}} < h_{i,C_{j_2}}$.

Ce qui signifie que la suite

$$u_j = (\sum_{1\leq i\leq n} k_{i,C_j}, \sum_{1\leq i\leq n} h_{i,C_j}) = (K_j, H_j)$$

est strictement croissante pour l'ordre lexicographique et pour $j < r$.

Par ailleurs, $H_j \leq \nu(K_j)$ par définitions de ρ, μ, ν. Il en résulte que, si $K_{j_1} = K_{j_2}$, alors $j_2 \leq j_1 + \nu(K_{j_1})$.

Par suite, $r \leq \sum_{1\leq q\leq K} \nu(q)$. D'où le résultat. □

On définit alors $\mathcal{F}' = \mathcal{F}_{\lambda_0}$ où λ_0 est la fonction constante qui vaut $\nu(K)$ pour tout non-terminal. (Nous ne cherchons pas ici à optimiser).

5.3.9 Correction et terminaison de l'algorithme de nettoyage

Rappelons tout d'abord l'algorithme de nettoyage:

1. $C := C_0$
2. Tant que $\mathcal{F}'(\mathcal{F}(C)) \neq \mathcal{F}'(C)$ et que $C(t_0, d_0) = \emptyset$
   $C := \mathcal{F}(C)$
3. Si $C(t_0, d_0) \neq \emptyset$ alors $[\![t_0, d_0]\!]$ n'est pas vide sinon $[\![t_0, d_0]\!]$ est vide.

Théoreme 5.33 *L'algorithme ci-dessus termine et est correct.*

Preuve

La terminaison résulte du lemme 5.27 (finitude de $\mathcal{F}_{\lambda_0}(C)(t, d)$ pour tout état C et tout (t, d)). En ce qui concerne la correction, dans le cas où $[\![t, d]\!]$ est vide le résultat est trivial puisqu'alors $C(t_0, d_0)$ est vide pour tout C. Intéressons nous donc au cas où $[\![t_0, d_0]\!]$ est non-vide. Comme $[\![t_0, d_0]\!] = \bigcup_{n\geq 0} \mathcal{F}^n(C_0)(t_0, d_0)$, il existe alors un nombre $r+1$ minimal tel que $\mathcal{F}^{r+1}(C_0)(t_0, d_0) \neq \emptyset$ et donc tel que $\mathcal{F}'(\mathcal{F}^r(C_0)) \neq \mathcal{F}^r(C_0)(t_0, d_0) = \emptyset$. Alors, d'après le lemme 5.32, $r \leq \nu(0) + \ldots + \nu(K)$. De plus, par construction de $\mathcal{F}'$ et d'après le lemme

5.29, pour tout $i \leq r$, il existe un non-terminal N_i tel que $\mathcal{F}'(\mathcal{F}^i(C_0))(N_i) \neq \mathcal{F}^i(C_0)(N_i)$. Par conséquent l'algorithme ne s'arrête pas avant que $C = \mathcal{F}^r(C_0)$.

Si bien que, lorsque $[\![t_0, d_0]\!] \neq \emptyset$, l'algorithme ne s'arrête que lorsqu'on a pu calculer un terme dans ce langage. □

5.3.10 Quelques remarques sur la méthode présentée ici

Nous avons présenté une preuve de la décidabilité de la réductibilité inductive, preuve différente de celle de D. Plaisted. Mais il faut remarquer que l'algorithme ci-dessus est, comme celui de D. Plaisted, inutilisable en pratique parce que nous n'avons pas essayé de donner ici la plus petite fonction $\mathcal{F}'$.

Nous pensons qu'il est possible de donner, sur la base de la méthode présentée ici, un algorithme beaucoup plus efficace. Ceci nous est suggéré par la simplicité des algorithmes de nettoyage dans le cas linéaire et dans le cas où l'ensemble des formes normales est fini (voir les exemples précédents) ou même dans d'autres cas particuliers décrits dans la section suivante. Au moins dans tous ces cas particuliers notre méthode s'avère efficace (de loin plus efficace que celle de Plaisted).

Admettant même que le problème soit intrinsèquement complexe (comme suggéré dans [KNZ86]), il semble plus facile, pour caractériser les problèmes conduisant à un test de réductibilité inductive simple, de se fonder sur la nature de la grammaire de formes normales, plutôt que sur le système de réécriture qui y conduit. (Par exemple, il existe des systèmes de réécriture non-linéaires tels que le langage de formes normales associé soit rationnel). De plus, comme nous l'avons déjà fait remarquer, un nettoyage de la grammaire de formes normales permet d'effectuer simultanément plusieurs tests de réductibilité inductive, ce qui permet de "factoriser" cette étape coûteuse.

Une méthode encore plus simple et efficace consiste à éviter les tests de réductibilité inductive. Ceci est possible par transformation de la spécification (transformation qui utilise précisément les grammaires de formes normales) comme il est montré dans [Com88a]. Mais la complexité du problème se retrouve alors dans la transformation de la spécification.

Du point de vue de la théorie des langages, nous obtenons un classe de langages pour laquelle le vide est décidable et qui n'est contenue dans aucune classe pour laquelle cette propriété est connue. On peut aussi noter que la classe des langages de formes normales est stable par union et intersection finie: $NF_{t,d} \cap NF_{t',d'}$ est calculé en résolvant le problème équationnel

$$\exists\, Var(t, t') : x = t \wedge x = t' \wedge d \wedge d'$$

Ses formes résolues sont de la forme $\exists \vec{w} : x = u \wedge \delta$. $NF_{t,d} \cap NF_{t',d'}$ est alors la réunion des $NF_{u,\delta}$ pour de telles formes résolues.

Mais nous n'avons pas étudié les langages pour eux-mêmes; il reste à étudier d'autres propriétés de stabilité, étudier la décidabilité de l'inclusion ou de l'égalité et à envisager des extensions de cette classe de langages.

5.4 Cas particuliers

De nombreux cas particuliers conduisent à des calculs bien plus simples que ceux qui sont exposés dans la section précédente. Un exemple de cas particulier très simple est celui où tous les termes de LHS sont linéaires puisque la grammaire de formes normales obtnue est rationnelle (si t est linéaire) ou (si t n'est pas linéaire) permet d'utiliser un algorithme de nettoyage "classique". Plus généralement, si aucune règle de la grammaire ne comporte de condition, l'algorithme de nettoyage classique (calcul de $\rightarrow_{\mathcal{G}}^{n}$ où n est le nombre de non-terminaux) s'applique. Cela peut arriver même lorsque LHS comporte des termes non-linéaires[17] comme le montre l'exemple:

Exemple 5.11 La signature est composée des symboles a (constante), h (unaire) et f (binaire). $LHS = \{h(f(x,x)), f(h(x_1),x_2), f(f(x_1,x_2),x_3)\}$ La grammaire calculée est alors de la forme:

$$\begin{array}{rcl} NF_{h(x)} & \rightarrow & h(NF_{h(x)} \\ & | & h(NF_{f(x_1,x_2),x_1\neq x_2} \\ NF_{f(x_1,x_2),x_1\neq x_2} & \rightarrow & f(NF_a, NF_{h(x)}) \\ & | & f(NF_a, NF_{f(x_1,x_2)}) \\ NF_{f(x_1,x_2)} & \rightarrow & f(NF_a, NF_x) \\ NF_a & \rightarrow & a \\ NF_x & \rightarrow & NF_a \\ & | & NF_{h(x)} \\ & | & NF_{f(x_1,x_2)} \end{array}$$

Noter que sur cet exemple, LHS comporte des termes non-linéaires et ceux-ci ne sont pas des instances d'autres termes de LHS. (Ce qui correspond à un cas trivial d'absence de condition).

Nous voulons montrer ici d'autres exemples (un peu plus difficiles) pour lesquels on obtient quand même un algorithme de nettoyage simple. Nous essaierons aussi de donner des indications pour l'amélioration de la méthode proposée dans la section précédente.

5.4.1 Cas où il n'y a "pas trop" de conditions dans la grammaire

Nous envisageons ici un cas restreint où, tout en autorisant LHS à contenir des termes non-linéaires, la grammaire obtenue ne contient "pas trop" de règles conditionnelles. L'idée est de formaliser la méthode esquissée sur l'exemple 5.10.

On définit un ordre partiel sur les non-terminaux de la façon suivante: $N < N'$ si

> Pour toute dérivation $A \Rightarrow_{\mathcal{G}}^{*} (U,C)$ telle que N' ait une occurrence dans U, l'une au moins des règles appliquées dans cette dérivation a pour membre gauche N. (A désigne ici l'axiome de la grammaire de formes normales $\mathcal{G}$)

$\leq$ est bien une relation d'ordre sur N si l'on suppose qu'il n'y a pas *d'inaccessibles* i.e. que pour tout non terminal N, il existe une dérivation $A \Rightarrow_{\mathcal{G}}^{*} (U,C)$ telle que N

[17] Un problème ouvert interessant est de caractériser les ensembles LHS qui conduisent à de telles grammaires.

ait une occurrence dans U. On peut supposer sans perdre de généralité qu'il n'y a pas d'inaccessibles dans la grammaire.

Remarquons que l'axiome A est plus petit que tous les autres non-terminaux.

Dans un premier temps supposons que la grammaire de formes normales $\mathcal{G}$ vérifie la condition suivante:

Condition 1

Pour toute règle R de $\mathcal{G}$ de la forme

$$NF_{t,d} \rightarrow v[NF_{t_1,d_1}, \ldots, NF_{t_n,d_n}] \text{ Si } \delta$$

où δ est une conjonction non-vide de diséquations, il existe un sous-ensemble $\mathcal{E}_R$ de $\{NF_{t_1,d_1}, \ldots, NF_{t_n,d_n}\}$ tel que

- $\forall z \neq u \in \delta$, $\exists NF_{t_i,d_i} \in \mathcal{E}_R$, $z \in Var(t_i)$ ou bien $Var(u) \subseteq Var(t_i)$
- $\forall NF_{t_i,d_i} \in \mathcal{E}_R$, ou bien $NF_{t_i,d_i} \leq NF_{t,d}$, ou bien les règles de grammaire dont le membre gauche est NF_{t_i,d_i} ne comportent pas de conditions.

Exemple 5.12 L'exemple 5.10 illustre bien la condition ci-dessus. Rappelons cet exemple:

$R1$	$NF_{x_1+x_2}$	$\rightarrow$	$NF_{s(x_1)} + NF_{s(x_2)}$ si $x_1 \neq x_2$
$R2$		$\mid$	$NF_{s(x_1)} + NF_{x_2+x_3}$
$R3$		$\mid$	$NF_{x_1+x_2} + NF_{s(x_3)}$
$R4$		$\mid$	$NF_{x_1+x_2} + NF_{x_3+x_4}$ si $x_1 \neq x_3$
$R5$		$\mid$	$NF_{x_1+x_2} + NF_{x_3+x_4}$ si $x_2 \neq x_4$
$R6$	$NF_{s(x)}$	$\rightarrow$	$s(NF_0)$
$R7$		$\mid$	$s(NF_{x_1+x_2})$
$R8$		$\mid$	$s(NF_{s(s(x))})$
$R9$		$\mid$	$s(NF_{s(x_1+x_2)})$
$R10$		$\mid$	$s(NF_{s(g(x))}$
$R11$	NF_0	$\rightarrow$	0
$R12$	$NF_{s(s(x))}$	$\rightarrow$	$s(NF_{s(s(x))})$
$R13$		$\mid$	$s(NF_{s(g(x))})$
$R14$		$\mid$	$s(NF_{s(x_1+x_2)})$
$R15$	$NF_{s(x_1+x_2)}$	$\rightarrow$	$s(NF_{x_1+x_2})$
$R16$	$NF_{s(g(x))}$	$\rightarrow$	λ

Les seules règles de la grammaire qui comportent des conditions ont pour membre gauche $NF_{x_1+x_2}$. Par conséquent, il suffit de choisir $\mathcal{E}_R = \emptyset$ pour toutes les règles qui n'ont pas $NF_{x_1+x_2}$ comme partie gauche et $\mathcal{E}_R$ égal à l'ensemble de tous les non-terminaux apparaissant dans R sinon. Pour tout élément NF_{t_i,d_i} de $\mathcal{E}_R$ on a alors en effet $NF_{t_i,d_i} = NF_{x_1+x_2}$ ou bien les règles de grammaire ayant NF_{t_i,d_i} comme membre gauche ne comportent pas de conditions.

Plus généralement, toutes les grammaires telles que un non-terminal seulement donne lieu à des règles conditionnelles vérifient la condition 1.

On associe alors à chaque non-terminal N un entier $\mathcal{M}(N)$, qui nous donnera le nombre maximum de termes à calculer dans $C(N)$. Ce nombre s'obtient ici de façon simple car

on peut effectivement assurer la satisfiabilité des conditions dans les règles de grammaire à l'aide de bornes sur le nombre d'éléments calculés dans chaque état.

Plus précisément, considérons par exemple la règle de grammaire

$$NF_{v[x_1,x_2]} \rightarrow v[NF_{t_1[y_1]}, NF_{t_2[y_2]}] \text{ si } y_1 \neq y_2$$

La condition 1 entraine (si $NF_{t_i[y_i]} \not\leq NF_{v[x_1,x_2]}$, ce que nous supposons ici) que les règles de grammaire dont les membres gauches sont $NF_{t_1[y_1]}$ ou $NF_{t_2[y_2]}$ ne comportent pas de condition. Elles sont donc de la forme

$$NF_{t_i[y_i]} \rightarrow t_i[NF_{w_i}]$$

Si l'on considère alors deux dérivations (ou réductions) consécutives, cela revient à utiliser la règle:

$$NF_{v[x_1,x_2]} \rightarrow v[t_1[NF_{w_1}], t_2[NF_{w_2}]] \text{ si } w_1 \neq w_2$$

La réduction par cette nouvelle règle étant garantie possible dès que NF_{w_1} ou NF_{w_2} possède au moins deux éléments (et que chacun d'eux possède au moins un élément). Plus précisément même, si l'on ajoute cette règle à la grammaire, on obtient:

$$|\mathcal{F}(\mathcal{F}(C))(NF_{v[x_1,x_2]})| \geq \max(|C(NF_{w_1})| * (|C(NF_{w_2})| - 1), |C(NF_{w_2})| * (|C(NF_{w_1})| - 1))$$

Ce genre d'inégalté directe permet de simplifier la définition de $\mathcal{F}'$.

Développons plus formellement cette idée. Pour commencer, construisons un "arbre des dérivations possibles" (ADP en abrégé). Cet arbre est un arbre ET/OU, chaque noeud OU étant étiqueté par un non-terminal N, ses successeurs sont les noeuds ET étiquetés par les règles de grammaire dont N est membre gauche. Les successeurs d'un noeud ET étiqueté par la règle $R : N \rightarrow v[N_1, \ldots, N_n]$ si δ sont des noeuds OU étiquetés respectivement par $N_1, \ldots, N_n$. La racine de l'ADP est un noeud OU étiqueté par l'axiome. Enfin, lorsqu'un noeud OU ou l'un de ses frères est étiqueté par un non-terminal étiquetant aussi un de ses ancêtres stricts, on ne développe pas ses fils.

La figure 5.5 illustre l'arbre des dérivations possibles de la grammaire de l'exemple 5.12.

Remarquons que l'ADP est toujours fini puisque chaque non-terminal ne peut apparaitre qu'au plus deux fois sur chaque chemin.

L'étape suivante consiste alors à associer, le long de chaque chemin, un nombre entier à chaque non-terminal, nombre qui représente le nombre de termes qu'il suffit de posséder dans un état du calcul de ce non-terminal pour assurer la possibilité d'une réduction le long de ce chemin.

Soyons plus précis. On parcourt l'ADP en associant à chaque noeud OU un entier : on associe 1 à la racine ainsi qu'à chaque position de taille 2. Lorsque ADP/p est associé à n et que ADP/$p \cdot i$ est une règle R dont la condition δ est de cardinal k, on associe à tout noeud de position $p \cdot i \cdot i_1 \cdot i_2 \cdot i_3$ ($i_1, i_2, i_3 \in \mathbf{N}$) le nombre $n * (k + 1)$. Cette situation est illustrée par la figure 5.6.

Ainsi, à chaque noueud OU de l'ADP est associé un entier .

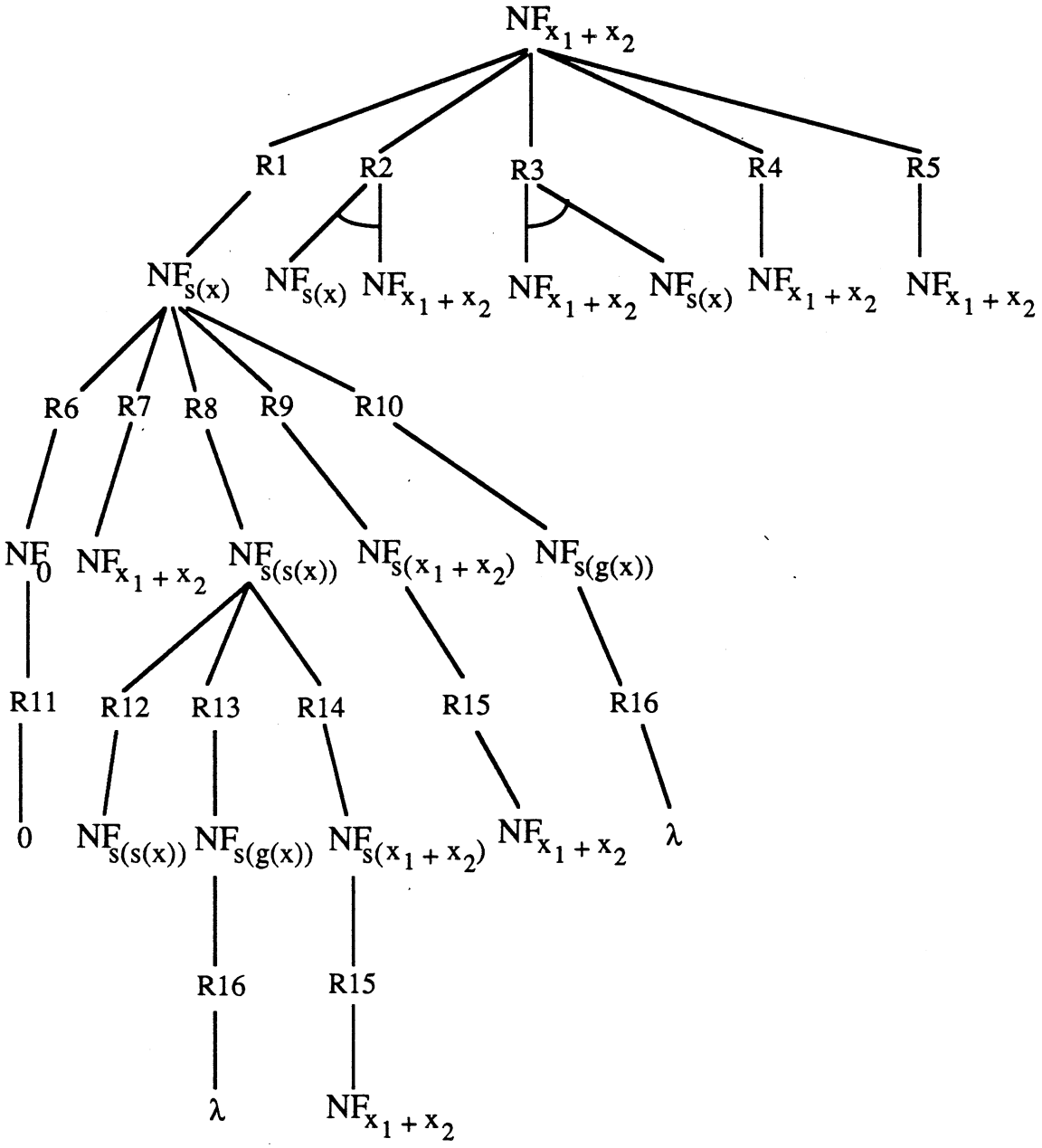

Figure 5.5: Un exemple d'arbre des dérivations possibles

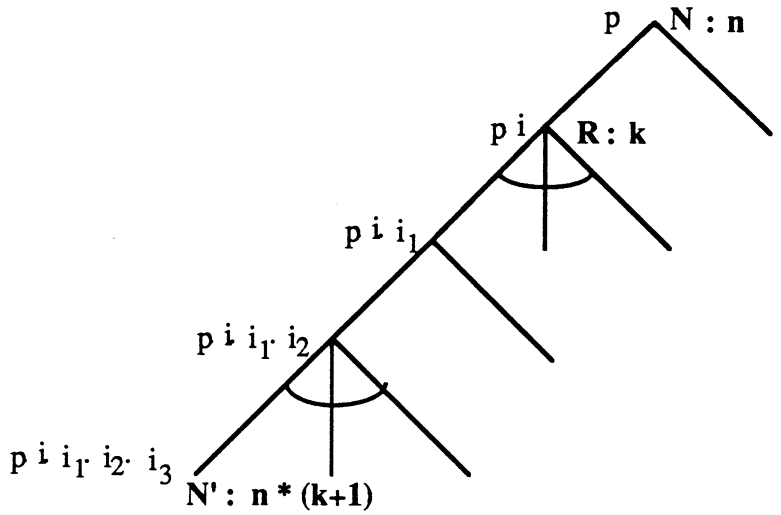

Figure 5.6:

Un *chemin* Γ de l'ADP est une suite de positions consécutives commençant par ϵ et ne contenant pas de feuille. Γ_{ou} est alors la suite des non-terminaux étiquetant les noeuds OU de Γ et Γ_{et} est la suite des règles étiquetant les noeuds ET de Γ. L'entier associé à chaque noeud définit une fonction (partielle) h qui associe à un non-terminal N et à un chemin Γ un entier $h(\Gamma, N)$. Cette fonction se prolonge en une application en associant 0 à tous les non terminaux qui n'apparaissent pas dans le chemin. On pose alors

$$h(N) = \max_\Gamma h(\Gamma, N)$$

pour tout non-terminal N (Γ parcourant l'ensemble des chemins de l'ADP). Par exemple, $h(A) = 1$ si A est l'axiome. Dans l'exemple 5.12 illustré par la figure 5.5, on retrouve les nombres inscrits dans les rectangles de la figure 5.3:

$$\begin{array}{ll} h(NF_{x_1+x_2}) = 1 & h(NF_{s(x)}) = 2 \\ h(NF_0) = 2 & h(NF_{s(x_1+x_2)}) = 2 \\ h(NF_{s(s(x))}) = 2 & h(NF_{s(g(x))}) = 2 \\ h(NF_{g(x)}) = 0 & h(NF) = 0 \end{array}$$

Lemme 5.34 *Soit* $d \equiv z_1 \neq u_1 \wedge \ldots \wedge z_n \neq u_n$ *tel que, pour tout* i, z_i *est une variable et* $z_i \notin Var(u_i)$. *Soit* $\{x_1, \ldots, x_m\} = Var(z_1, \ldots, z_m, u_1, \ldots, u_m)$ *et* $A_1, \ldots, A_m \subseteq T(F)$. *Soit enfin* $\alpha \in \mathbf{N}$.

Si, pour tout i, $|A_i| \geq n + \alpha$, *alors* d *possède au moins* $\alpha * (n + \alpha)^m$ *solutions* σ *telles que, pour tout* i, $x_i\sigma \in A_i$.

Preuve

On raisonne par récurrence sur m: si $m = 1$, alors u_1 est un terme fermé et l'ensemble des substitutions σ telles que $x_1\sigma \in A_1$ et $x_1\sigma \notin u_1$ est de cardinal supérieur à $|A_1| - 1$ c'est à dire supérieur à $\alpha * (\alpha + 1)^0$.

Supposons maintenant la propriété vraie pour $m-1$. ($m \geq 2$). On découpe d en deux morceaux: $d \equiv d_1 \wedge d_2$ où d_1 ne contient pas d'occurrence de x_m et toutes les diséquations de d_2 contiennent au moins une occurrence de x_m. Soit k le nombre de diséquations dans d_1. Par hypothèse de récurrence, d_1 possède au moins $(\alpha+n-k)*(\alpha+n)^{m-2}$ solutions σ telles que, $\forall i < m, x_i\sigma \in A_i$. Si σ est maintenant une telle solution de d_1, $d_2\sigma$ ne contient pas de diséquation triviale. Par décompositions, l'une des formes irréductibles de $d_2\sigma$ est $\top$ ou bien de la forme $x_m \neq t_1 \wedge \ldots \wedge x_m \neq t_k$ où $t_1, \ldots, t_k$ sont des termes fermés[18]. Par conséquent, A_m étant de cardinal $\alpha+k+n-k$, il existe au moins $\alpha+k$ substitutions θ qui sont solutions de $d_2\sigma$ et telles que $x_m\theta \in A_m$.

Il en résulte qu'il existe au moins $(\alpha+k)*(\alpha+n-k)*(\alpha+n)^{m-2}$ substitutions $\sigma\theta$ qui sont solutions de d et telles que, pour tout i, $x_i\sigma\theta \in A_i$.

Or un calcul élémentaire de minimum montre que la fonction qui à k associe $(\alpha+k)*(\alpha+n-k)$ atteint son minimum sur l'intervalle $[1,n]$ en $k=n$. D'où le résultat. □

Lemme 5.35 *Soit p une position de l'ADP telle que $ADP(p) = N \in NT$. Soit $ADP(p \cdot i) = R$ une règle de $\mathcal{G}$ de condition δ telle que $n = |\delta| > 0$. Soient ensuite $NF_{t_1,d_1}, \ldots, NF_{t_k,d_k}$ les fils de R et C un état du calcul de $\mathcal{G}$. Soit enfin $\alpha \in \mathbf{N}$.*

Si, pour tout $i \in \{1,\ldots k\}$, il existe une règle R_i étiquetant un fils de N_i telle que, pour tout fils $N_{i,j}$ de R_i, $|C(N_{i,j})| \geq n+\alpha$

Alors

$$|\mathcal{F}^2(C)(N)| \geq \alpha*(n+\alpha)^m$$

où $m = \sum_{i=1}^{k} |Var(t_i)|$.

Preuve

C'est une conséquence du lemme précédent (et des hypothèses sur la grammaire) : si $A_{i,j} = C(N_{i,j})$, d'après le lemme précédent, δ possède au moins $(\alpha+n)^{m-1}*\alpha$ solutions σ telles que, pour toute variable $x_{i,j} \in Var(t_i)$, $x_{i,j}\sigma \in A_{i,j}$. Mais alors, les règles R_i étant inconditionnelles, pour chacune de ces substitutions σ et pour tout i, $t_i\sigma \in \mathcal{F}(C)(t_i,d_i)$ et $t\sigma \in \mathcal{F}^2(C)(N)$. □

Lemme 5.36 *Soit Γ un chemin de l'ADP et $p \in \Gamma_{ou}$. Si $ADP(p) = N$ et si $|\mathcal{F}^2(C)(N)| < h(\Gamma,N)$, alors $\forall i \in \mathbf{N}, \exists j \in \mathbf{N}, \forall k \in \mathbf{N}, \exists l \in \mathbf{N}$,*

$$p \cdot i \cdot j \cdot k \cdot l \in Pos(ADP) \Rightarrow |C(ADP(p \cdot i \cdot j \cdot k \cdot l))| < h(ADP(p \cdot i \cdot j \cdot k \cdot l))$$

Preuve

Prouvons la contraposée du lemme: On suppose qu'il existe une règle R telle que, pour tout fils N_i de R, existe une règle R_i telle que, pour tout fils $N_{i,j}$ de R_i, $|C(Ni.j)| > h(N_{i,j})$. D'aprè le lemme 5.35, on a alors

$$|\mathcal{F}^2(C)(N)| \geq (h(N_{i,j})-n)*h(N_{i,j})^{m-1}$$

[18]Il n'est pas possible que toutes les formes irréductibles soient $\perp$ car, d'après le lemme 3.7 $d_2\sigma$ possède au moins une solution dans $T(F)$.

Mais, par définition, $h(N_{i,j}) \geq (n+1) * h(\Gamma, N)$. Donc

$$|\mathcal{F}^2(C)(N)| \geq (n * h(\Gamma, N) + h(\Gamma, N) - n) * ((n+1) * h(\Gamma, N))^{m-1}$$

Si l'on note alors que $h(\Gamma, N) > 0$, le résultat du lemme est une conséquence immédiate de cette inégalité. □

Ce lemme est fondamental car il permet de limiter le calcul des états à des états de cardinal borné à l'aide de h:

Le calcul restreint $\mathcal{F}'$ est ici défini par:

- si $|C(N)| < h(N)$ alors $\mathcal{F}'(C)(N) = \mathcal{F}(C)(N)$
- sinon, $\mathcal{F}'(C)(N) = C(N)$

L'algorithme ressemblera alors au précédent (mais avec un calcul restreint beaucoup plus simple):

$C := C_0$
Tant que $\mathcal{F}'^2(C) \neq C$ **et** $C(A) = \emptyset$ **faire** $C := \mathcal{F}'^2(C)$
$C(A) = \emptyset$ **si et seulement si** $\mathcal{L}(A) = \emptyset$

Cet algorithme possède bien entendu les propriétés voulues, comme nous allons le voir.

Lemme 5.37 *Soit $d \equiv z_1 \neq u_1 \wedge \ldots \wedge z_n \neq u$ tel que, pour tout i, z_i est une variable et $z_i \notin Var(u_i)$. Soit $\{x_1, \ldots, x_m\} = Var(z_1, \ldots, z_n, u_1, \ldots, u_n)$. Soient $A_1, \ldots, A_m$ m sous-ensembles de $T(F)$. Soit α un entier.*

Supposons que A_1 est de cardinal supérieur ou égal à $n+\alpha$ et que d possède β solutions σ telles que, pour tout i, $x_i\sigma \in A_i$. Alors $\beta < \alpha \Rightarrow \beta = 0$.

Preuve
Ce résultat est essentiellement une reformulation du lemme 5.23 dans un cas particulier.

Le lemme qui suit établit qu'un nouveau terme dans $\mathcal{F}'^2(C)(N)$ ou bien "provient" d'un terme calculé lui aussi par $\mathcal{F}'$ ou bien ne peut contribuer à la solution; nous dirons qu'un non-terminal N est *inutile* (dans l'état C) si, pour tout chemin Γ de l'ADP tel qu'il existe une position p avec $ADP(p) = N$, il existe une position $q \in \Gamma$ telle que $q <_{pref} p$, $ADP(q) = N_0$ et $|\mathcal{F}^2(C)(N_0)| \geq h(N_0)$.

Lemme 5.38 *Si N est un non terminal tel que $\mathcal{F}'^2(\mathcal{F}^2(C))(N) \neq \mathcal{F}^2(C)(N)$ alors*

- *ou bien il existe un non terminal N' tel que $\mathcal{F}'^2(C)(N') \neq C(N')$ et N' n'est pas inutile*
- *ou bien N est inutile*

Preuve
Supposons que nous ne sommes pas dans le deuxième cas énoncé dans le lemme. Soit donc un chemin Γ et une position p de Γ tels que, pour toute position q de Γ précédant p, $|\mathcal{F}^2(C)(ADP(q))| < h(ADP(q))$. Deux cas se présentent alors :

- ou bien il existe deux positions p, q de Γ telles que $ADP(p) = N_0$, $ADP(q) = N$, $p <_{pref} q$ et $\mathcal{F}'^2(C)(N_0) \neq C(N_0)$ et le résultat est trivial
- ou bien tout ancêtre N_0 de N (dans Γ) vérifie $\mathcal{F}'^2(C)(N_0) = C(N_0)$. Mais comme, par hypothèse, $|\mathcal{F}^2(C)(N_0)| < h(N_0)$, $\mathcal{F}'^2(C)(N_0) = \mathcal{F}^2(C)(N_0)$ et donc $\mathcal{F}^2(C)(N_0) = C(N_0)$.

  Soir R une règle permettant de produire $t \in \mathcal{F}'^2(\mathcal{F}^2(C))(N) - \mathcal{F}^2(C)(N)$. R étiquette un noeud de l'ADP à cause de la propriété que nous venons de voir. Il existe alors un fils N_1 de R tel qu'il existe un $u \in \mathcal{F}'(\mathcal{F}^2(C))(N_1) - \mathcal{F}(C)(N_1)$. Soit R_1 une règle permettant de produire u et telle qu'il existe un fils N_2 de R_1 et un terme $v \in \mathcal{F}^2(C)(N_2) - C(N_2)$. La suite N, R, N_1, R_1, N_2 est nécessairement une suite d'étiquettes de positions consécutives de l'ADP. Et, d'après les lemmes 5.36 et 5.37, $|C(N_2)| < h(N_2)$, d'où $\mathcal{F}'^2(N_2) \neq C(N_2)$.

□

Des lemmes qui précèdent on déduit le résultat souhaité :

Théoreme 5.39 *L'algorithme de décision du vide termine et est correct.*

Preuve

La seule difficulté est de prouver que, si, pour tout N, $\mathcal{F}'^2(C)(N) = C(N)$ et que $C(A) = \emptyset$, alors $\mathcal{L}(A) = \emptyset$. Plaçons nous donc dans ce cas. Par récurrence sur n, à l'aide du lemme 5.38 on obtient que $\mathcal{F}'^2(\mathcal{F}^n(C))(N) \neq \mathcal{F}^n(C)(N)$ entraine que N est inutile (dans $\mathcal{F}^n(C)$). En particulier, comme il n'y a pas d'état dans lequel A soit inutile, $\mathcal{F}'^2(\mathcal{F}^n(C))(A) = \mathcal{F}^n(C)(A)$ pour tout n. Donc (comme $C(A) = \emptyset$), $\mathcal{F}^n(C)(A) = \emptyset$ pour tout n. D'où le résultat.□

5.4.2 Un lemme technique et son application au cas de deux variables

Nous donnons ici un nouveau résultat technique, espérant pouvoir le généraliser pour obtenir des algorithmes plus efficaces.

Lemme 5.40 *Soit $NF_{t,\delta} \rightarrow t[NF_{t_1,d_1}, \ldots, NF_{t_n,d_n}]$ si d une règle de $\mathcal{G}$. Soit $\{y_1, \ldots, y_m\} = \bigcup_{i>1} Var(t_i)$. Soit p_1 une position de t telle que $t[NF_{t_1,d_1}, \ldots, NF_{t_n,d_n}]/p_1 \equiv NF_{t_1,d_1}$. On suppose que*

$$C(NF_{t_1,d_1}) \neq \{u/p_1 \mid u \in \mathcal{F}(C)(NF_{t,\delta})\}$$

Alors, il existe un nombre $k_1 \leq |d|$, m nombres entiers $l_1, \ldots, l_m$ et m sous-ensembles $E_1, \ldots, E_m$ de $T(F)$ tels que

- $l_1 + \ldots + l_m = k_1$
- *pour tout i, $|E_i| = l_i$*
- *Pour toutes substitutions $(\sigma_2, \ldots, \sigma_n) \in C(t_2, d_2) \times \ldots \times C(t_n, d_n)$, il existe un i tel que $y_i\sigma_2 \ldots \sigma_n \in E_i$.*

Ce résultat généralise en fait le lemme 5.30 (voir aussi la figure 5.4) en raffinant les conclusions que l'on peut tirer lorsqu'un élément du langage $[\![t_1, d_1]\!]$ est "stoppé" par la condition d d'une règle de grammaire. Le résultat est particulièrement intéressant lorsque $m = 2$. En effet,

- ou bien aucune valeur n'est "stoppée" et tout fonctionne comme s'il n'y avait pas de condition dans la règle de grammaire
- ou bien, pour chaque substitution sur t_1 qui est stoppée, on peut limiter les valeurs des variables de t_2 à des "bandes" (cf figure 5.4). Dans ce dernier cas, cela a, bien sûr, un impact sur le nombre de valeurs à calculer puisque l'on ne peut pas avoir beaucoup de substitutions dépendantes...

Nous ne détaillons pas ici, en espérant que cette explication très approximative suffira au lecteur pour se faire une idée.

Preuve

On raisonne par récurrence sur n. Pour $n = 1$, il n'y a pas de diséquations et donc

$$C(NF_{t_1,d_1}) = \{u/p_1 \mid u \in \mathcal{F}(C)(NF_{t,\delta})\}$$

Supposons maintenant que $\phi \in C(t_1, d_1) - \{u/p_1 \mid u \in \mathcal{F}(C)(NF_{t,\delta})\}$. Aucune substitution $\theta = \sigma_2 \ldots \sigma_n$ avec $(\sigma_2, \ldots, \sigma_n) \in C(t_2, d_2) \times \ldots \times C(t_n, d_n)$ n'est alors solution de $d\phi$.

Décomposons alors d en $d \equiv D \wedge d'$ où d' est constitué des diséquations $u \neq v$ de d telles que u et v contiennent chacun au moins une occurrence de variable qui n'est pas dans $Var(t_1)$ (et dans ce cas, $Var(t_1) \cap Var(u, v) = \emptyset$ par propriété des problèmes simples.) On note encore D_1 et d'_1 les diséquations obtenues par application de ϕ. Si bien que D_1 est une conjonction $u_1 \neq v_1 \wedge \ldots \wedge u_r \neq v_r$ où $u_1, \ldots, u_r$ sont des termes fermés. Soit D_2 une des formes irréductibles de D_1 obtenue par décomposition et élimination des disjonctions: une diséquation de D_2 a pour membre gauche une variable et pour membre droit un terme fermé. Par hypothèse sur ϕ, aucune des substitutions θ de la forme $\theta = \sigma_2 \ldots \sigma_n$ où $\sigma_i \in C(t_i)$ n'est solution de $D_2 \wedge d'_1$.

Pour chaque variable y_i, soit H_i l'ensemble (éventuellement vide) des termes fermés w tels que $y_i \neq w$ soit une diséquation de D_2. Deux cas se présentent alors:

- ou bien, pour tout θ, il existe un indice i tel que $y_i\theta \in H_i$, et le lemme est prouvé
- ou bien il existe des substitutions $\sigma_2, \ldots, \sigma_n$ dans $C(t_2, d_2) \times \ldots \times C(t_n, d_n)$ telles que, pour tout indice i, $y_i\sigma_2 \ldots \sigma_n \notin H_i$.

Supposons donc que nous nous trouvons dans ce dernier cas et considérons l'état C' défini par:

$$C'(NF_{t_i,d_i}) = C(NF_{t_i,d_i}) - \{\sigma \in \Sigma_g \mid y_j\sigma \in H_j\}$$

$\forall \sigma_2, \ldots, \sigma_n \in C'(t_2, d_2) \times \ldots \times C'(t_n, d_n), \sigma_2 \ldots \sigma_n$ n'est pas solution de d'_1 (sinon, $\phi\sigma_2 \ldots \sigma_n \in \mathcal{F}(C)(t, \delta)$). De plus, cet ensemble est non vide par hypothèse.

On peut dire alors, par exemple, qu'il existe une substitution σ_2 dans $C(t_2, d_2)$ telle que,

$$\forall \sigma_3, \ldots, \sigma_n \in C'(t_3, d_3) \times \ldots \times C'(t_n, d_n),$$

$(t[NF_{t_1,d_1},\ldots,NF_{t_n,d_n}],\phi\sigma_2\ldots\sigma_n)$ ne se réduit pas par $\rightarrow_{\mathcal{G}}$.

On applique alors l'hypothèse de récurrence à la règle

$$NF_{t\phi,d\phi} \rightarrow t\phi[NF_{t_2,d_2},\ldots,NF_{t_n,d_n}] \text{ si } d'_1$$

Il existe donc un nombre $k_2 \leq |d'_1|$, $l'_1,\ldots,l'_{m'},E'_1,\ldots,E'_{m'}$ tels que $l'_1+\ldots+l'_m = k_2$, $\forall i, |E'_i| = l'_i$ et

$$\forall\sigma_3,\ldots,\sigma_n \in C'(t_3,d_3)\times\ldots\times C'(t_n,d_n),\ \exists i(>2),\ y_i\sigma_3\ldots\sigma_n \in E'_i$$

Cette dernière propriété entrainant bien entendu que

$$\forall\sigma_2,\ldots,\sigma_n \in C'(t_2,d_2)\times\ldots\times C'(t_n,d_n),\ \exists i(>2),\ y_i\sigma_2\ldots\sigma_n \in E'_i$$

Il suffit alors de choisir $k_1 = k_2 + \sum|H_j|$, $l_i = |H_i| + l'_i$ et $E_i = H_i \cup E'_i$. La seule propriété restant à vérifier est que $k_1 \leq |d|$. Pour cela, il suffit de remarquer que $|d'_1| + |D_2| \leq |d|$ et $|D_2| = \sum|H_j|$, par définition de H_j. □

Pour conclure, il faut avouer qu'il reste à faire dans la direction indiquée ci-dessus. Même les résultats de la section précédente mériteraient un approfondissement afin d'une part de dégager clairement "ce qui fait marcher" la méthode et d'autre part de clarifier et d'optimiser les résultats qui y sont donnés. Les résultats développés dans ce chapitre ont surtout pour mérite de proposer une alternative à la méthode de Plaisted.

Chapitre 6

Autres applications

Il ne s'agit pas ici de tenter de donner une liste exhaustive des autres applications des problèmes équationnels, mais d'en donner un aperçu. Nous avons déjà vu dans le chapitre précédent comment utiliser la simplification des problèmes de équationnels pour calculer des grammaires de formes normales. Il se trouve que la simplification des problèmes équationnels est utilisée plus ou moins explicitement dans divers domaines. Rappelons en quelques-uns:

- L'analyse temporelle des processus communicants en FP2 [Sch87b]
- La compilation du filtrage et ses applications en programmation fonctionnelle [Sch88a]
- La programmation logique avec contraintes [JL87]
- L'unification AC [Bur88]
- Les transformations de programmes logiques [Lug88]
- La négation explicite en programmation logique: il s'agit de rendre explicite l'information négative contenue dans un programme logique lorsque l'on fait l'hypothèse du monde clos (cf [BMPT87], [Lug88]).
- La représentation explicite de termes définis par contre-exemples [LM87].
- etc ...

Tous ces travaux ont en commun -d'une part l'utilisation plus ou moins explicite de la simplification des problèmes équationnels comprenant des paramètres -d'autre part de ne s'intéresser qu'aux calculs sur les termes fermés.

Nous allons ici mettre en évidence les outils utilisés pour ces applications et indiquer d'autres directions pour leur utilisation. Dans un premier temps, nous montrons comment utiliser la simplification des problèmes équationnels pour effectuer des opérations sur les forêts quadrillées. Nous montrons ensuite comment rendre explicite l'information négative contenue dans un programme logique constitué de clauses de Horn (voir [Lug88] pour certaines extensions aux programmes généraux). Enfin, nous montrons qu'il n'est pas possible

de spécifier l'égalité sur les entier relatifs sans opérateur caché et nous ébauchons une application du langage des formes normales fermées à la transformation de spécifications (voir [Com88a] pour plus de détails).

6.1 Opérations sur les forêts quadrillées

6.1.1 Forêts quadrillées

L'idée présente, par exemple en FP2 [Jor86], est que l'on peut représenter un ensemble infini de termes fermés (ensemble d'états d'un processus en FP2) par un ensemble fini de termes avec variables. Malheureusement, la famille de langages d'arbres ainsi définie n'est pas stable par complémentaire, ce qui serait pourtant utile pour certaines applications [Sch87b]. Nous avons par ailleurs déjà vu dans les chapitres précédents une famille de langages d'arbres plus expressive en introduisant les termes contraints. C'est dans cette direction que nous allons définir les "forêts quadrillées" qui constituent une famille de langages d'arbres stable par toutes les opérations usuelles. En fait, les forêts quadrillées constituent un cas particulier des langages de formes normales. Mais il est utile de les étudier séparément car les opérations sur ces langages s'effectuent de manière particulièrement simple.

(S, F) est une signature fixée une fois pour toute. On supposera $T(F)$ non vide. Rappelons la définition d'un terme contraint.

Définition 6.1 *Un* terme contraint *est une paire (t, d) où $t \in T(F, X)$ et d est une conjonction de diséquations de la forme $z \neq u$ où z est une variable (finitaire) de t et u est un sous-terme linéaire de t ne contenant pas z.*

Nous n'aurons en fait pas besoin ici des hypothèses u sous-terme de t et u linéaire. La définition d'un terme contraint peut donc être généralisée en omettant ces deux conditions. Les résultats proposés ici restent valides.

Comme dans le chapitre précédent, si (t, d) est un terme contraint, $[\![t, d]\!]$ désigne l'ensemble des instances fermées de t qui sont solution de d. Les résultats du chapitre 3 assurent en particulier que $[\![t, d]\!]$ n'est jamais vide.

Définition 6.2 *Une* forêt quadrillée *est un sous-ensemble $\mathcal{Q}$ de $T(F)$ pour lequel il existe un ensemble fini de termes contraints $(t_1, d_1), \ldots, (t_n, d_n)$ tels que*

$$\mathcal{Q} = \bigcup_{i=1}^{n} [\![t_i, d_i]\!]$$

6.1.2 Opérations sur les ensembles infinis de termes

Outre la réunion finie, le complémentaire et l'intersection finie, on souhaite pouvoir calculer (un ensemble T de termes étant donné ainsi qu'un système de réécriture $\mathcal{R}$) l'ensemble des termes qui sont obtenus à partir de T par réécriture en tête en une étape (cf [Sch87b]). Ces opérations sont utiles en FP2, langage dans lequel les règles de réécriture sont utilisées non pas pour la simplification mais pour exprimer des transitions d'états. Un terme fermé

représente en effet un état d'un système de transition, un terme avec variable représentant l'ensemble des états qui en sont des instances fermées. Un règle de réécriture représente ainsi un ensemble infini de règles de transitions d'états.

D'une façon générale, nous voulons pouvoir calculer l'ensemble des termes en relation avec les termes d'un ensemble donné T, lorsque cette relation est *finiment présentée.*

Définition 6.3 *Une relation R définie sur $T(F)$ est dite* finiment présentée *s'il existe*

1. *un ensemble fini de paires $\{((t_1,u_1),c_1),\ldots,((t_n,u_n),c_n)\}$ où pour tout i $t_i,u_i \in T(F,X)$ et c_i est une conjonction de diséquations dont les variables sont contenues dans $Var(t_i,u_i)$*
2. *Un ensemble de positions P fini*

tels que tRu ssi

$$\exists i \in \{1,\ldots,n\},\ \exists \sigma \in \mathcal{S}(T(F),c_i,Var(t_i,u_i)),\ \exists p \in P,\ t/p \equiv t_i\sigma,\ u/p \equiv u_i\sigma$$

Par exemple, lorsque tous les c_i sont égaux à $\top$ (i.e. intuitivement, il n'y a pas de conditions dans la réécriture), et lorsque $P = \{\epsilon\}$, on retrouve la réécriture en tête[1].

On définit alors les opérations suivantes sur les ensembles de termes fermés: si $T \subseteq T(F)$ et R est une relation finiment présentée sur $T(F)$,

- $Succ_R(T) = \{t \in T(F) \mid \exists t' \in T, t'Rt\}$
- $Pred_R(T) = \{t \in T(F) \mid \exists t' \in T, tRt'\}$

6.1.3 Stabilité des forêts quadrillées

Théoreme 6.4 *Si R est une relation finiment présentée, l'ensemble $\mathcal{F}_Q$ des forêts quadrillées est stable pour les opérations intersection finie, union finie, complémentaire, $Succ_R$, $Pred_R$.*

Preuve

La preuve que nous donnons est de plus constructive : nous montrons comment calculer l'image d'une forêt quadrillée par les différentes opérations énoncées dans le théorème.

Réunion finie La stabilité par réunion finie est triviale: il suffit de considérer la réunion des termes contraints qui définissent chacune des forêts

Intersection finie Il suffit de montrer comment construire l'intersection de $[\![t_1,d_1]\!]$ et de $[\![t_2,d_2]\!]$. On peut supposer sans perdre de généralité que t_1 et t_2 sont de même sorte (sinon l'intersection est vide) et que $Var(t_1) \cap Var(t_2) = \emptyset$ (Ce qu'on obtient au prix d'un renommage).

Soient $\mathcal{P}_1,\ldots,\mathcal{P}_n$ les formes irréductibles[2] du problème équationnel $\exists Var(t_1,t_2)$: $x = t_1 \wedge x = t_2 \wedge d_1 \wedge d_2$ d'inconnue principale x (qui est supposé ne pas appartenir à

[1] Noter que la relation de réduction associé à un système de réécriture en général n'est pas une relation finiment présentée. Les calculs de $Succ_R$ et $Pred_R$ s'avèrent en effet impossibles dans ce cas (ou tout du moins difficiles).

[2] Pour $\mathcal{R}_2$ par exemple.

$Var(t_1, t_2)$. Pour tout i, $\mathcal{P}_i \equiv \exists \vec{w_i} : x = u_i \wedge c_i$. Alors (par correction et complétude de l'algorithme de réduction des problèmes équationnels),

$$[\![t_1, d_1]\!] \cap [\![t_2, d_2]\!] = \bigcup_{i=1}^{n} [\![u_i, c_i]\!]$$

Passage au complémentaire Comme nous avons vu comment calculer l'intersection finie de forêts quadrillées et que le complémentaire d'une réunion finie est une intersection finie de complémentaires, il nous suffit de montrer comment calculer le complémentaire de $[\![t, d]\!]$.

Soient $\mathcal{P}_1, \ldots, \mathcal{P}_n$ les formes irréductibles du problème $\forall Var(t) : x = t \vee \neg d$. (d'inconnue principale x. Si $\mathcal{P}_i \equiv \exists \vec{w_i} : x = u_i \wedge d_i$ on obtient alors (la preuve est triviale)

$$T(F) - [\![t, d]\!] = \bigcup_{i=1}^{n} [\![u_i, d_i]\!]$$

$Succ_R$ Supposons que R est défini par P et $((t_1, u_1), c_1), \ldots, ((t_n, u_n), c_n)$. Comme

$$Succ_R(\cup_i T_i) = \cup_i Succ_R(T_i)$$

on peut, là encore, se limiter au calcul de $Succ_R$ et $Pred_R$ sur $[\![t, d]\!]$. On supposera de plus que les positions de P -ou bien sont des positions de t -ou bien ne sont préfixées par aucune position variable de t. (On peut se ramener à ce cas d'après le lemme 6.5 ci-dessous). On suppose aussi sans perdre de généralité que $Var(t_1, \ldots, t_n, u_1, \ldots, u_n) \cap Var(t) = \emptyset$. Pour chaque position $p \in P \cap Pos(t)$ et chaque indice $i \in \{1, \ldots, n\}$, soit $\mathcal{P}_{1,p,i}, \ldots, \mathcal{P}_{k_{p,i},p,i}$ l'ensemble des formes irréductibles du problème

$$\exists Var(u_i, t, t_i) : x = u_i \wedge t/p = t_i \wedge c_i \wedge d$$

Si $\mathcal{P}_{j,p,i} \equiv \exists \vec{w_{j,p,i}} : x = v_{j,p,i} \wedge d_{j,p,i}$, alors

$$Succ_R([\![t, d]\!]) = \bigcup_{p \in P \cap Pos(t), 1 \leq i \leq n, 1 \leq j \leq k_{p,i}} [\![t[v_{j,p,i}]_p, d_{j,p,i}]\!]$$

(C'est une conséquence directe des définitions)

$Pred_R$ Il suffit de remarquer que $Pred_R = Succ_{R^{-1}}$ où R^{-1} est la relation obtenue en échangeant u_i et t_i pour tout i. □

Lemme 6.5 *Soit P un ensemble fini de positions et (t, d) un terme contraint. Alors il existe un nombre fini de termes contraints $(t_1, d_1), \ldots, (t_n, d_n)$ tels que $[\![t, d]\!] = \bigcup_{i=1}^{n} [\![t_i, d_i]\!]$ et tels que, pour tout i et toute position variable p de t_i, p n'est pas préfixe d'une position de P.*

Preuve

Soit x une variable de t dont une position dans t est un préfixe d'une position de P. Pour

tout $f \in F$, soit $\mathcal{P}_f$ le problème $\exists Var(t), \exists \vec{w} : x' = t \wedge x = f(\vec{w}) \wedge d$. Soient $\mathcal{P}_{f,1}, \ldots, \mathcal{P}_{f,k(f)}$ les formes irréductibles de $\mathcal{P}_f$. Soit $\mathcal{P}_{f,j} \equiv \exists \vec{w'} : x' = u_{f,j} \wedge d_{f,j}$ Alors

$$[\![t, d]\!] = \bigcup_{f \in F, 1 \leq i \leq k(f)} [\![u_{f,j}, d_{f,j}]\!]$$

Il suffit alors de répéter cette transformation tant que c'est possible. Elle n'est possible qu'un nombre fini de fois (et donc il y a terminaison) en effet: soit, pour chaque position variable p de t qui est un préfixe d'une position de P, $n(p)$ le maximum de $\{|q| - |p| \; | q \in P \; p \leq q\}$. Le multi-ensemble des nombres $n(p)$ décroit à chaque application de la transformation.□

6.2 Négation explicite en programmation logique

Ici, comme dans la section précédente, notre ambition n'est pas de donner des résultats révolutionnaires mais d'esquisser certaines applications. C'est pourquoi nous ne considérons ici que des programmes écrits sous forme de clause de Horn même s'il est probable que les résultats puissent être étendus.

Un programme peut donc s'écrire comme un ensemble de clauses[3] de la forme:

$$P(t_1, \ldots, t_n) \leftarrow Q_1(u_{1,1}, \ldots, u_{1,m_1}) \wedge \ldots \wedge Q_k(u_{k,1}, \ldots, u_{k,m_k})$$

où P, $Q_1, \ldots, Q_k$ sont des littéraux et $t_1, \ldots, t_n, u_{1,1}, \ldots, u_{1,m_1}, \ldots, u_{k,1}, \ldots, u_{k,m_k}$ sont des termes de $T(F, X)$[4].

De façon opérationnelle, si l'on cherche à trouver les instances $G\sigma$ d'un but G (qui est une conjonction de littéraux positifs), qui sont valides dans la classe des modèles d'un programme $\mathcal{P}$, il suffit (et il faut) de trouver toutes les substitutions σ telles que $\mathcal{P} \cup \{\leftarrow G\sigma\}$ n'ait aucun modèle. On connait plusieurs procédures de semi-décision d'un tel problème (qui est indécidable en général). Par exemple la fameuse SLD-résolution [Llo84]. Ces procédures consistent en un ensemble fini de règles d'inférence (par exemple le modus ponens) qui transforment un ensemble de clauses en un ensemble de clauses ayant mêmes modèles et d'un stratégie *complète*. Partant ainsi d'un programme $\mathcal{P} \cup \{\leftarrow G\}$, est construit un arbre (appelé arbre de dérivation) dont chaque noeud est un programme et tel que les successeurs d'un noeud N sont tous les programmes qui peuvent être obtenus à partir de N en employant une des règles d'inférence en accord avec la stratégie. (En fait, en l'absence de règle destructrice, les noeuds de l'arbre de dérivation sont plutôt les clauses *ajoutées* par l'étape d'inférence). Les feuilles d'un tel arbre sont alors -soit un ensemble de clauses contenant la clause vide (on dira qu'il y a succès) -soit un ensemble de clauses ne contenant pas la clause vide et pour lesquelles aucune règle ne s'applique (on dira qu'il y a échec). Un chemin de l'arbre de dérivation conduisant à la clause vide correspond à une preuve de la validité de la formule $\exists x_1, \ldots, x_n, G[x_1, \ldots, x_n]$. La preuve construisant

[3] Nous ne rappelerons pas ici toutes les définitions (c'est hors du sujet de cette thèse). Nous utiliserons les définitions de [Llo84].

[4] Remarquons que nous supposons ici que chaque clause contient *exactement* un littéral positif. Autrement dit, le programme ne contient pas de but.

explicitement des valeurs pour $x_1, \ldots, x_n$, c'est à dire une substitution σ dont le domaine est l'ensemble des variables de G. La stratégie est complète si toute substitution σ telle que $G\sigma$ est une conséquence logique de $\mathcal{P}$, il existe une substitution θ, plus générale que σ qui est calculée le long d'un chemin de l'arbre de dérivation conduisant à la clause vide.

Ces procédures ne sont que des procédures de semi-décision en ce sens que *si*

$$\exists x_1, \ldots, x_n, G[x_1, \ldots, x_n]$$

est valide alors la procédure permet de le prouver en temps fini (mais non borné).

Il en résulte que, lorsque $\exists x_1, \ldots, x_n, G[x_1, \ldots, x_n]$ n'est pas valide, l'éxécution du programme ne termine pas. Pour éviter cet inconvénient dans certains cas, on introduit la règle de *négation par échec*. Cette règle (que nous noterons NCE) consiste à inférer $\forall x_1, \ldots, x_n, \neg G[x_1, \ldots, x_n]$ lorsque l'arbre de dérivation est fini et ne contient que des échecs. Bien sûr, lorsque l'arbre de dérivation est infini et ne contient que des échecs, l'éxécution du programme logique ne terminera pas.

La règle NCE n'est pas valide dans tous les modèles du programme, mais seulement dans *le plus petit modèle de Herbrand* (dans le cas des clauses de Horn). Afin d'étudier cette règle, différentes notions ont été introduites. Rappelons celles que nous allons utiliser ici.

Comme dans [Cla78],il est possible d'associer à chaque symbole de prédicat n-aire n variables distinctes de celles qui apparaissent dans le programme (et distinctes de celles qui sont associées aux autres symboles de prédicat), toute clause de la forme

$$P(t_1, \ldots, t_n) \leftarrow Q_1(u_{1,1}, \ldots, u_{1,m_1}) \wedge \ldots \wedge Q_k(u_{k,1}, \ldots, u_{k,m_k})$$

étant transformée en :

$$P(x_1, \ldots, x_n) \leftarrow Q_1(u_{1,1}, \ldots, u_{1,m_1}) \wedge \ldots \wedge Q_k(u_{k,1}, \ldots, u_{k,m_k}) \wedge x_1 = t_1 \wedge \ldots \wedge x_n = t_n$$

Il est courant en programmation logique de faire *l'hypothèse du monde clos* [Rei78]. Cette hypothèse consiste à affirmer que tout ce qui n'est pas une conséquence logique du programme est faux. Cette hypothèse est valide (dans le cas des clauses de Horn) dans le plus petit modèle de Herbrand du programme, généralement choisi pour décrire la sémantique déclarative [Llo84].

Autrement dit, si $P(x_1, \ldots, x_n) \leftarrow C_1, \ldots, P(x_1, \ldots, x_n) \leftarrow C_m$ sont les clauses du programme dont P est le littéral positif (rappelons que nous avons effectué la transformation ci-dessus), on a, dans le plus petit modèle de Herbrand:

$$P(x_1, \ldots, x_n) \leftrightarrow C_1 \vee \ldots \vee C_m$$

C'est la notion de "programme complété" de Clark [Cla78] (restreinte au cas des clauses de Horn).

On peut remarquer que, dans l'exposé succint que nous avons fait ci-dessus de l'éxécution d'un programme logique, les littéraux positifs et négatifs ne jouent pas des rôles identiques.

En particulier, les buts ne peuvent être que de la forme $\exists x_1, \ldots, x_n, G$ ou $\forall x_1, \ldots, x_n \neg G$ (dans le cas où l'on utilise la règle NCE) où G est un littéral positif. On ne peut donc ni traiter des buts de la forme $\exists x_1, \ldots, x_n, \neg G$ ni des buts de la forme $\forall x_1, \ldots, x_n, G$. Notre objectif dans cette section est d'indiquer comment, sous l'hypothèse du monde clos, compléter un programme logique en ajoutant des clauses de Horn permettant de "définir" des prédicats correspondant à la négation de ceux qui sont définis dans le programme initial. Cela permet de traiter les buts négatifs de la même manière que les buts positifs. Cette symétrie permet en particulier d'augmenter la puissance de déduction des méthodes évoquées ci-dessus puisque le cas des formules $\exists x_1, \ldots, x_n, \neg G$ peut être traité, de même que celui des formules $\forall x_1, \ldots, x_n, G$ (en utilisant la négation par échec).

Montrons cela sur un exemple.

Exemple 6.1 Supposons que S ne contient qu'une sorte et que F ne contient que deux symboles : 0 (constante) et s (unaire). PAIR est alors défini par:

$$\begin{aligned} &\mathrm{PAIR}(0) \leftarrow \\ &\mathrm{PAIR}(s(s(x))) \leftarrow \mathrm{PAIR}(x) \end{aligned}$$

Notons maintenant IMPAIR le prédicat $\neg$PAIR. Avec l'hypotrhèse du monde clos, nous pouvons écrire (comme rappelé plus haut) :

$$\mathrm{PAIR}(x) \leftrightarrow x = 0 \vee (\exists y, x = s(s(y)) \wedge \mathrm{PAIR}(y))$$

En prenant la négation de cette équivalence, on obtient:

$$\mathrm{IMPAIR}(x) \leftrightarrow x \neq 0 \wedge (\forall y, x \neq s(s(y)) \vee \mathrm{IMPAIR}(y))$$

Les résultats des chapitres 3 et 4 fournissent alors des outils pour transformer cette formulation en un ensemble de clauses:

$$\begin{aligned} &\mathrm{IMPAIR}(s(0)) \leftarrow \\ &\mathrm{IMPAIR}(s(s(x))) \leftarrow \mathrm{IMPAIR}(x) \end{aligned}$$

Ce qui permet, par exemple, de prouver la validité (dans le plus petit modèle de Herbrand) de la formule $\exists x, \mathrm{IMPAIR}(x)$. Formule dont on ne pouvait prouver la validité par les moyens classiques.

Une transformation analogue est exposée dans [BMPT87].

Avant d'exposer plus en détail la méthode, il nous faut généraliser un peu la notion de problème équationnel pour pouvoir traiter les symboles de prédicats.

6.2.1 Généralisation des problèmes équationnels

Etant donnée une signature (S, F) et un ensemble de variables X, $T(F, X)$ désigne comme précédemment l'algèbre des termes bien formés construite sur F et X. Nous considérerons désormais aussi un ensemble de symboles de prédicats $\mathcal{L}$ et l'ensemble $\widetilde{\mathcal{L}} = \{\widetilde{P} \mid P \in \mathcal{L}\}$. L'intention est que $\widetilde{P}$ coincide avec $\neg P$. Pour l'instant ce sont seulement de nouveaux symboles de prédicat.

A chaque symbole de prédicat est associé une arité (comme pour les symboles fonctionnels) qui est un élément de S^* (et non plus de $S^* \times S$). Un *littéral* est une expression $P(t_1, \ldots, t_n)$ où $P \in \mathcal{L} \cup \widetilde{\mathcal{L}}$ a pour arité $\underline{s}_1 \ldots \underline{s}_n$ et, pour tout i, t_i est un terme de $T(F, X)$ de sorte $\underline{s}_i$.

Définition 6.6 *Un* système généralisé *est*

- *ou bien un littéral*
- *ou bien une équation ou une diséquation*
- *ou bien une conjonction de systèmes*
- *ou bien une disjonction de systèmes*

Définition 6.7 *Un* problème équationnel généralisé *est une formule*

$$\exists \vec{w}, \forall \vec{y} : B$$

où B est un système généralisé.

La syntaxe des problèmes équationnels est ainsi étendue pour inclure les littéraux. Bien entendu, tous les axiomes équationnels de l'algèbre des problèmes équationnels sont étendus. (cf figure 2.1).

Il ne reste qu'à définir la sémantique. (C'est celle que l'on attend !)

Définition 6.8 *Un* modèle *de $F, \mathcal{L} \cup \widetilde{\mathcal{L}}$ est un couple $(\mathcal{A}, I)$ où $\mathcal{A}$ est une F-algèbre et I une fonction (dite d'interprétation) qui à tout symbole $P \in \mathcal{L} \cup \widetilde{\mathcal{L}}$ d'arité $\underline{s}_1 \ldots \underline{s}_n$ associe une partie*[5] *de $\mathcal{A}_{\underline{s}_1} \times \ldots \times \mathcal{A}_{\underline{s}_n}$.*

Définition 6.9 *Soit $M = (\mathcal{A}, I)$ un modèle de $F, \mathcal{L} \cup \widetilde{\mathcal{L}}$. Une $\mathcal{A}$-substitution σ est une* solution dans *M du système $\mathcal{P}$ si*

- *ou bien $\mathcal{P}$ est une équation $u = v$ et $u\sigma =_{\mathcal{A}} v\sigma$*
- *ou bien $\mathcal{P}$ est une diséquation $u \neq v$ et $u\sigma \neq_{\mathcal{A}} v\sigma$*
- *ou bien $\mathcal{P} \equiv \top$*
- *ou bien $\mathcal{P}$ est un littéral $Q(t_1, \ldots, t_n)$ avec $n > 0$ et $(t_1\sigma, \ldots, t_n\sigma) \in I(Q)$*
- *ou bien $\mathcal{P}$ est un littéral d'arité 0 et $I(P) = true$*
- *ou bien $\mathcal{P}$ est une expression $P_1 \wedge \ldots \wedge P_n$ et σ est une solution de tous les P_i*
- *ou bien $\mathcal{P}$ est une expression $P_1 \vee \ldots \vee P_n$ et σ est solution de l'un des P_i*

Définition 6.10 *Soit $M = (\mathcal{A}, I)$ un modèle de $F, \mathcal{L} \cup \widetilde{\mathcal{L}}$. Une $\mathcal{A}$-substitution σ est une* solution dans *M du problème équationnel généralisé $\mathcal{P} \equiv \vec{w}, \forall \vec{y} : B$, par rapport à l'ensemble d'inconnues principales $\mathcal{I} \supseteq VL(\mathcal{P})$ si*

- *$Dom(\sigma) = \mathcal{I}$*

[5] Par convention, si $n = 0$, I associe à P soit le symbole *true* soit le symbole *fals*.

- *il existe une $\mathcal{A}$-substitution θ de domaine $\vec{w}$ telle que, pour toute $\mathcal{A}$-substitution ϕ de domaine $\vec{y}$, $\sigma\theta\phi$ est une solution dans M de B.*

Les notions de règles de transformations, correction, adéquation, forte adéquation se généralisent aux problèmes équationnels généralisés de manière évidente. Lorsque le modèle M n'est pas précisé, c'est que la règle considérée est correcte (resp. adéquate, resp. fortement adéquate) pour tout modèle.

Les résultats de correction, adéquation, forte adéquation du chapitre 3 sont en fait aussi valides pour les problèmes équationnels généralisés. De même les résultats de terminaison restent valides (mais bien sûr il n'en est pas de même des résultats de complétude). En fait, nous ne considérerons dans la suite que le cas $\mathcal{A} = T(F)$ et utiliserons donc les résultats s'y rapportant sans mention particulière.

6.2.2 Calcul de la contrepartie négative d'un programme

Un programme $\mathcal{P}$ est un ensemble de clauses de Horn dont les symboles de prédicats sont dans $\mathcal{L}$. Nous supposons tout d'abord que le programme $\mathcal{P}$ a été réécrit de sorte que toutes ses clauses contiennent des variables distinctes. (Ce qui est possible par renommage). Puis le programme est transformé comme indiqué plus haut de façon à ce que toutes ses clauses soient de la forme

$$P(x_1,\ldots,x_n) \leftarrow Q_1(u_{1,1},\ldots,u_{1,m_1}) \wedge \ldots \wedge Q_k(u_{k,1},\ldots,u_{k,m_k}) \wedge x_1 = t_1 \wedge \ldots \wedge x_n = t_n$$

où $x_1,\ldots,x_n$ sont des variables distinctes telles que toutes les têtes de clauses où apparaissent le même littéral soient identiques et que deux têtes de clauses où apparaissent des littéraux distincts n'aient pas de variable en commun.

Soient alors, pour tout symbole $P \in \mathcal{L}$, C_P l'ensemble des corps des clauses dont la tête est $P(x_1,\ldots,x_n)$. Comme rappelé plus haut, la formule $P \leftrightarrow \bigvee_{C \in C_P} C$ est valide dans le plus petit modèle de Herbrand de $\mathcal{P}$. Afin de déduire un programme définissant $\neg P$, on considère la négation de cette formule. La formule $\neg \bigvee_{C \in C_P} C$ est transformée en utilisant les règles:

$$\begin{array}{rcl} \neg(A \vee B) & \rightarrow & \neg A \wedge \neg B \\ \neg(A \wedge B) & \rightarrow & \neg A \vee \neg B \\ \neg Q(t_1,\ldots,t_k) & \rightarrow & \tilde{Q}(t_1,\ldots,t_k) \\ \neg(x = t) & \rightarrow & x \neq t \end{array}$$

On obtient ainsi pour chaque symbole $P \in \mathcal{L}$ une formule $\mathcal{F}_P$. Soient alors $\{y_1,\ldots,y_r\}$ les variables de $\mathcal{F}_P$ qui sont distinctes de $x_1,\ldots,x_n$ et soient $\mathcal{Q}_1,\ldots\mathcal{Q}_k$ les formes irréductibles du problème équationnel généralisé

$$\mathcal{Q} \equiv \forall y_1,\ldots,y_r : \mathcal{F}_P$$

Ces formes irréductibles ont les propriétés suivantes (la vérification est laissée au lecteur):

- $\mathcal{Q}_1,\ldots,\mathcal{Q}_k$ ne contiennent pas de paramètres si pour toute clause $H \leftarrow B$ de $\mathcal{P}$, $Var(B) \subseteq Var(H)$
- Il n'y a aucune diséquation dans $\mathcal{Q}_1,\ldots,\mathcal{Q}_k$ si toutes les têtes de clauses de $\mathcal{P}$ sont linéaires.

Lorsque ces deux conditions sont remplies, $\mathcal{Q}_i$ est de la forme:

$$\mathcal{Q}_i \equiv x_1 = u_1 \wedge \ldots \wedge x_n = u_n \wedge \widetilde{Q_1}(\ldots) \wedge \ldots \wedge \widetilde{Q_s}(\ldots)$$

On complète alors le programme $\mathcal{P}$ en ajoutant les clauses

$$\widetilde{P}(u_1, \ldots, u_n) \leftarrow \widetilde{Q_1}(\ldots) \wedge \ldots \wedge \widetilde{Q_s}(\ldots)$$

On note $\widetilde{\mathcal{P}}$ l'ensemble de clauses ainsi ajoutées à $\mathcal{P}$.

Théoreme 6.11 *Soit $\mathcal{P}$ un programme dans lequel toutes les têtes de clauses sont linéaires et tel que les variables des corps des clauses soient aussi des variables de la tête de clause correspondante. L'algorithme décrit ci-dessus calcule alors un programme $\mathcal{P} \cup \widetilde{\mathcal{P}}$ tel que, pour tout $P \in \mathcal{L}$, pour tous termes fermés $t_1, \ldots, t_n$, $P(t_1, \ldots, t_n) \wedge \widetilde{P}(t_1, \ldots, t_n)$ est faux dans le plus petit modèle de Herbrand de $\mathcal{P}$.*

Preuve

Remarquons tout d'abord que, par construction de $\widetilde{\mathcal{P}}$ et par correction des transformations de problèmes équationnels généralisés, $\widetilde{P}(t_1, \ldots, t_n)$ est vrai dans le plus petit modèle de Herbrand ssi $\{x_1 \rightarrow t_1, \ldots, x_n \rightarrow t_n\}$ est une solution de $\mathcal{Q}$ (nous reprenons ici les notations utilisées pour décrire l'algorithme). Le plus petit modèle de Herbrand est, par définition, le plus petit point fixe de $T_{\mathcal{P}}$ (cf [Llo84] par exemple; $T_{\mathcal{P}}$ associe à toute interprétation de Herbrand I l'interprétation contenant tous les termes de que l'on peut obtenir en une étape à l'aide d'une clause de $\mathcal{P}$ et de termes déjà contenus dans I). Grâce à des résultats de continuité, le plus petit modèle de Herbrand est classiquement la limite de la suite $T_{\mathcal{P}} \uparrow i$ (cf [Llo84]).

Nous allons donc prouver que, pour tout symbole de prédicat P, pour tous termes fermés $t_1, \ldots, t_n$ tels que $P(t_1, \ldots, t_n) \in T_{\mathcal{P}} \uparrow \omega$, alors $\widetilde{P}(t_1, \ldots, t_n) \notin T_{\mathcal{P} \cup \widetilde{\mathcal{P}}} \uparrow \omega$.

Pour cela, raisonnons par récurrence sur N tel que $P(t_1, \ldots, t_n) \in T_{\mathcal{P}} \uparrow N$:

- S'il existe un axiome $P(u_1, \ldots, u_n) \leftarrow$ dans $\mathcal{P}$ et une substitution θ telle que, pour tout i, $u_i\theta \equiv t_i$, alors, par définition des solutions d'un problème équationnel généralisé, $\{x_1 \rightarrow t_1; \ldots; x_n \rightarrow t_n\}$ n'est pas solution de $\forall Var(u_1, \ldots, u_n) : x1 \neq u_1 \vee \ldots \vee x_n \neq u_n$. Il en résulte que σ n'est pas solution de $\forall y_1, \ldots, y_r : \mathcal{Q}$ et donc que $\widetilde{P}(t_1, \ldots, t_n)$ est faux.

- Si $P(t_1, \ldots, t_n) \in T_{\mathcal{P}} \uparrow N$ et $P(t_1, \ldots, t_n) \notin T_{\mathcal{P}} \uparrow N - 1$ alors, par définition de $T_{\mathcal{P}}$, $P(t_1, \ldots, t_n)$ est obtenu par application d'une clause $P(u_1, \ldots, u_n) \leftarrow Q_1(\ldots) \wedge \ldots \wedge Q_s(\ldots)$ avec la substitution θ telle que $Q_1(\ldots)\theta, \ldots, Q_s(\ldots)\theta \in T_{\mathcal{P}} \uparrow N - 1$. Alors, par hypothèse de récurrence, $\widetilde{Q_1}(\ldots)\theta, \ldots, \widetilde{Q_s}(\ldots)\theta$ sont faux dans le plus petit modèle de Herbrand. D'autre part, l'une des formules de C_P est $x_1 = u_1 \wedge \ldots \wedge x_n = u_n \wedge Q_1(\ldots) \wedge \ldots \wedge Q_s(\ldots)$. Si $\sigma = \{x_1 \rightarrow t_1; \ldots x_n \rightarrow t_n\}$, alors $\sigma\theta$ n'est pas solution de

  $$\forall Var(u_1, \ldots u_n) : x_1 \neq u_1 \vee \ldots \vee x_n \neq u_n \vee \widetilde{Q_1}(\ldots) \vee \ldots \widetilde{Q_s}(\ldots)$$

  et n'est donc pas solution de $\mathcal{Q}$. Il en résulte que $\widetilde{P}(t_1, \ldots, t_n)$ est faux dans le plus petit modèle de Herbrand. □

Remarques et conjectures

1. Il faut noter que $\tilde{\mathcal{P}}$ ne modifie pas les interprétations des prédicats définis dans $\mathcal{P}$: il n'y a pas d"'interférence" entre $\mathcal{P}$ et $\tilde{\mathcal{P}}$.

2. Il se peut que, même dans le plus petit modèle de Herbrand, $P(t_1,\ldots,t_n)\vee\tilde{P}(t_1,\ldots,t_n)$ ne soit pas toujours vrai. Par exemple, si P est défini par le programme:

$$P(x) \leftarrow P(s(x))$$

   Alors, l'algorithme décrit plus haut calculera la contrepartie négative:

$$\tilde{P}(x) \leftarrow \tilde{P}(s(x))$$

   Les prédicats P et $\tilde{P}$ sont alors faux pour tout terme fermé dans le plus petit modèle de Herbrand.

   Cela prouve qu'en général $\tilde{P}$ n'est qu'une "approximation" de $\neg P$. Le théorème ci-dessus prouvant la cohérence de cette approximation.

3. La propriété symétrique de celle du théorème 6.11 semble elle aussi correcte:

   > Pour tous termes fermés $t_1,:\ldots,t_n$, $P(t_1,\ldots,t_n)\vee\tilde{P}(t-1,\ldots,t_n)$ est vrai dans le plus grand modèle de Herbrand.

   Faire en effet le même raisonnement que pour le théorème 6.11, en notant que le plus grand modèle de Herbrand est aussi le plus grand point fixe de $T_{\mathcal{P}}$ et s'obtient comme limite de $T^n_{\mathcal{P}}(I_0)$ où I_0 est l'interprétation dans laquelle tout est vrai.

4. Nous conjecturons aussi que, pour tout $P\in\mathcal{L}$, $\tilde{\tilde{P}} = P$ dans le plus petit modèle de Herbrand. Cette conjecture est vraisemblable puisque $\tilde{P}\subseteq\neg P$ d'où $P\subseteq\neg\tilde{P}$ et d'autre part $\tilde{\tilde{P}}\subseteq\neg\tilde{P}$.

6.2.3 Extensions

On peut envisager trois types d'extensions au théorème 6.11:

1. Comment étendre la méthode proposée au cas où certaines têtes de clauses ne sont pas linéaires?

2. Comment traiter le cas où certaines variables apparaissent dans le corps d'un clause sans apparaitre dans la tête correspondante?

3. Comment étendre la méthode à des programmes logiques autres que des clauses de Horn?

Ces trois hypothèses nous ont en effet été utiles dans l'énoncé de l'algorithme ou dans la preuve du théorème 6.11. L'étude de ces extensions sort du cadre de cette thèse. Examinons néanmoins rapidement les deux premières extensions.

Cas des têtes de clause non-linéaires

Dans ce cas il peut subsister des diséquations dans les formes résolues de $\mathcal{Q}$. Une possibilité pour contourner cette difficulté est d'introduire un prédicat DIFF qu'il faut aussi définir. Ce qui ne pose pas de problèmes dans les modèles libres; il suffit en effet d'écrire:

$$\begin{array}{lll} \mathrm{DIFF}(f(x_1,\ldots,x_n),g(y_1,\ldots,y_k)) & \leftarrow & \text{Pour tous } f,g \in F, f \neq g \\ \mathrm{DIFF}(f(x_1,\ldots,x_n),f(y_1,\ldots,y_n)) & \leftarrow \mathrm{DIFF}(x_i,y_i) & \text{Pour tout } i \end{array}$$

Dans tous les modèles de ce programme, si $t \neq_M t'$, alors $\mathrm{DIFF}(t,t')$ est vrai. Par contre, la réciproque est fausse et deux termes peuvent être égaux dans M alors que $\mathrm{DIFF}(t,t')$ est vrai.

Cas où certaines variables apparaissent dans un corps de clause sans apparaitre dans la tête

Dans ce cas, il n'est pas possible d'éliminer complètement les paramètres en simplifiant $\mathcal{F}_P$. Par contre, les paramètres qui subsistent apparaissent nécessairement au sein d'un littéral de la forme $Q(t_1,\ldots,t_n)$.

En fait, chaque problème irréductible $\mathcal{Q}_i$ est de la forme:

$$\mathcal{Q}_i \equiv \exists w_1,\ldots,w_m,\forall y_1,\ldots,yq : x_1 = t_1 \wedge \ldots \wedge x_n = t_n \wedge \delta_1 \wedge \ldots \wedge \delta_n$$

où chaque δ_i est de la forme $\widetilde{Q_1}(u_{1,1},\ldots,u_{1,n_1}) \vee \ldots \vee \widetilde{Q_s}(u_{s,1},\ldots,u_{s,n_s})$.

L'idée (que nous ne développerons pas formellement) est de traiter ces cas en utilisant la règle de négation par échec (NCE) sur les paramètres qui subsistent.

Par exemple, supposons que $\mathcal{Q}_1 \equiv \exists w,\forall y : x_1 = 0 \wedge x_2 = s(w) \wedge \widetilde{Q}(w,y)$. Si σ est une substitution de domaine $\{w\}$ et que l'arbre de preuve de $Q(w\sigma,y)$ est fini et ne contient que des échecs, il est possible d'inférer $\widetilde{P}(0,s(w)\sigma)$.

Autrement dit, on procède comme dans le paragraphe précédent, mais en ajoutant une règle spéciale pour les paramètres qui n'ont pas été éliminés : $\mathcal{Q}_1$ est transformé en la clause:

$$\widetilde{P}(0,s(w)) \leftarrow \mathrm{NCE}_{y,Q}(w,y)$$

Précisons un tout petit peu la construction sémantique de telles transformations:

Pour chaque δ_j, on introduit un symbole de prédicat R_j défini par l'unique clause:

$$R_j(y_1,\ldots,y_q,w_1,\ldots,w_m) \leftarrow Q_1(u_{1,1},\ldots,u_{1,n_1}) \wedge \ldots \wedge Q_s(u_{s,1},\ldots,u_{s,n_s})$$

si

$$\delta_j \equiv \widetilde{Q_1}(u_{1,1},\ldots,u_{1,n_1}) \vee \ldots \vee \widetilde{Q_s}(u_{s,1},\ldots,u_{s,n_s})$$

Pour chaque $\mathcal{Q}_i$, on ajoute alors la clause:

$$\widetilde{P}(t_1,\ldots,t_n) \leftarrow \bigwedge_j \mathrm{NCE}_{y_1,\ldots,y_q,R_j}(y_1,\ldots,y_q,w_1,\ldots,w_m)$$

$\mathrm{NCE}_{y_1,\ldots,y_q,P}(\vec{z})$ ayant pour ensemble de succès les substitutions fermées σ dont le domaine est $\vec{z} - \{y_1,\ldots,y_q\}$ et telle que $P(\vec{z}\sigma)$ possède un arbre de résolution fini ne contenant que des échecs.

Nous ne précisons pas plus, et ne donnons pas de résultats à ce sujet. Ce thème est développé par Barbutti & all. dans un article qui paraitra sans doute dans "Journal of Logic Programming". On peut aussi consulter [Lug88].

6.3 Transformations de spécifications

Nous allons montrer ici que, même dans un cas très simple de spécification algébrique multi-sortes, il n'est pas possible de définir correctement l'égalité. Or, dans une spécification avec sortes ordonnées ce problème devient aisé à résoudre (cf [GM87b]). Nous indiquerons alors (sans donner un développement complet) comment l'on peut transformer automatiquement une spécification multi-sorte en une spécification avec sortes ordonnées équivalente (dans le sens où les théorèmes inductifs sont "conservés"). Cette transformation permet, modulo le fait que l'on sache effectuer des preuves par consistence dans les théories avec sortes ordonnées, de ramener le problème des preuves par induction dans les théories équationnelles au cas où il n'y a pas de relations entre constructeurs (cf section 5.1.3). Et ainsi d'éviter les tests de réductibilité inductive qui constituent la partie la plus pénible de l'algorithme de Jouannaud et Kounalis [JK86b].

6.3.1 L'égalité ne peut être spécifiée correctement sur les entiers relatifs

Reprenons les définitions du chapitre 5: nous nous intéressons aux spécifications algébriques *hiérarchiques* et nous souhaitons que toute nouvelle définition vienne s'ajouter aux précédentes sans les modifier. Plus précisément, nous nous intéressons à la *protection* des spécifications algébriques telle qu'elle est définie dans le chapitre 5. Reprenons en d'ailleurs un des exemples:

Exemple 6.2 $S = \{\underline{int}, \underline{bool}\}$. $F = \{true, false : \rightarrow \underline{bool};\ 0 : \rightarrow \underline{int};\ s, p : \underline{int} \rightarrow \underline{int};\ eq : \underline{int} \times \underline{int} \rightarrow \underline{bool}\}$. $E = \{s(p(x)) == x;\ p(s(x)) == x\} \cup E_1$ où E_1 est un ensemble d'équations dont un des membres a pour racine eq.

Nous allons prouver qu'il n'existe pas d'ensemble fini E_1 tel que pour tous termes $t, u \in T(F)$ de sorte $\underline{int}$, $t =_E u \Leftrightarrow eq(t,u) =_E true$. Cela entraine en particulier les deux paradoxes suivants:

- si l'on a les deux propriétés $t =_E u \Rightarrow eq(t,u) =_E true$ et $t \neq_E u \Rightarrow eq(t,u) =_E false$ alors $true =_E false$ (i.e. la spécification est incohérente).
- si l'on a les deux propriétés $eq(t,u) =_E true \Rightarrow t =_E u$ et $eq(t,u) =_E false \Rightarrow t \neq_E u$ alors il existe un terme de sorte $\underline{bool}$ qui n'est ni $true$ ni $false$ (i.e. la spécification n'est pas suffisamment complète)

Ceci mettra en évidence l'insuffisance des spécifications multi-sorte en prouvant qu'il n'est pas possible de spécifier l'égalité sur les entiers relatifs sans introduire d'opérateur caché.

Les équations de E seront toujours considérées de façon symétrique[6]. i.e. si $u == v \in E$, nous supposerons que $v == u \in E$. T_1 désignera l'ensemble des termes de $T(F)$ de sorte $\underline{int}$. De plus, comme nous ne considérerons pas dans la suite de variable de sorte $\underline{bool}$, toutes les variables seront implicitement de sorte $\underline{int}$.

Donnons maintenant quelques définitions permettant de classer les spécifications de eq:

[6] Cela revient soit à considérer que $==$ est commutatif soit que E a été complété.

Définition 6.12
- E_1 *est dit* trivial *si* $\forall u, v, u', v' \in T_1,\ eq(u,v) =_E eq(u',v')$
- E_1 *est dit* symétrique *si* $\forall u, v \in T_1,\ eq(u,v) =_E eq(v,u)$
- E_1 *est dit* complet *si* $\forall u, v \in T_1,\ eq(u,v) =_E true$ *ou* $eq(u,v) =_E false$

Enfin ϕ est l'unique homomorphisme de T_1 dans l'ensemble des entiers relatifs $\mathbf{Z}$. (s est interprété comme le successeur $s_{\mathbf{Z}}$ et p comme le prédécesseur $p_{\mathbf{Z}}$). ϕ se prolonge de façon unique aux termes de sorte *int* de $T(F,X)$ en associant 0 à toute variable (de sorte *int*).

Lemme 6.13 *Soit u un terme de sorte int tel que $Var(u) = \{x\}$. Alors*

1. $\forall \sigma \in \Sigma,\ \phi(u\sigma) =_{\mathbf{Z}} \phi(u) + \phi(x\sigma)$
2. $\forall u_0 \in T(F,X),\ u_0$ *de sorte int*, $\exists \sigma \in \Sigma,\ u\sigma =_E u_0$
3. $\forall u, v \in T_1,\ (u =_E v \Leftrightarrow \phi(u) =_{\mathbf{Z}} \phi(v))$

Preuve

La propriété 1 se prouve par récurrence sur la profondeur de u : c'est une conséquence du fait que ϕ est un morphisme.

Si $u_0 \in T(F,X)$ est de sort *int*, sa forme normale pour le ystème de réécriture (canonique) $\mathcal{R} = \{s(p(x)) \to x;\ p(s(x)) \to x\}$ est de la forme $s^n(y)$ ou $p^n(y)$ ou 0. De même, $u\downarrow_{\mathcal{R}}$ est de l'une des formes $s^k(x), p^k(x), 0$. Il est alors facile de vérifier, pour chacun des 9 cas possibles pour le couple $(u\downarrow, u_0\downarrow)$ qu'il existe une substitution σ sur x telle que $u\downarrow \sigma =_E u_0\downarrow$. D'où la propriété 2.

Lorsque $u, v \in T_1,\ u =_E v \Leftrightarrow u\downarrow_{\mathcal{R}} \equiv v\downarrow_{\mathcal{R}}$. Or, si $u\downarrow \equiv v\downarrow$, alors $\phi(u\downarrow) =_{\mathbf{Z}} \phi(v\downarrow)$. Comme d'autre part $u_1 \to_{\mathcal{R}} u_2 \Rightarrow \phi(u_1) =_{\mathbf{Z}} \phi(u_2)$, on en déduit que $\phi(u\downarrow) =_{\mathbf{Z}} \phi(v\downarrow) \Rightarrow \phi(u) = \phi(v)$. D'où la propriété 3. □

Lemme 6.14 *Si E_1 est non trivial et symétrique et que $eq(u,v) == w \in E_1$ avec $|Var(u,v)| = 2$, alors $Var(w) = Var(u,v)$.*

Preuve

remarquons tout d'abord qu'un terme de $T(F,X)$ contient toujours au plus deux variables (de sorte *int*). Ainsi, si $|Var(u,v)| = 2$, alors $Var(u) = \{x\}$ et $Var(v) = \{y\}$ avec $x \neq y$. Supposons, par exemple, que $x \notin Var(w)$. Alors, pour tout $v' \in T_1$, soit $\sigma_{v'}$ la substitution associant à y et à toute variable de w le terme v'. Pour toute substitution θ de domaine $\{x\}$, $eq(u\theta, v\sigma_{v'}) =_E w\sigma_{v'}$. Par conséquent, d'après la propriété 2 du lemme 6.13, pour tout terme $u' \in T_1$, $eq(u', v\sigma_{v'}) =_E w\sigma_{v'}$. Comme, de plus, v' est quelconque, on peut énoncer (avec une nouvelle application du lemme 6.13):

$$\forall v' \in T_1, \exists t_{v'} \in T_1, \forall u' \in T_1, eq(u', v') =_E t_{v'}$$

Montrons maintenant que $t_{v'}$ est indépendant de v'. soient v' et v'' deux termes fermés distincts. $\forall u' \in T_1, eq(u', v') =_E t_{v'}$ et $\forall u'' \in T_1, eq(u'', v'') =_E t_{v''}$. En particulier si l'on

choisit $u'' \equiv v'$ et $v'' \equiv u'$. On trouve alors $eq(v'', v') =_E eq(v', v'')$. Par suite, $t_{v''} =_E t_{v'}$. On peut alors en déduire que

$$\exists t \in T_1, \forall v' \in T_1, \forall u' \in T_1, eq(u', v') =_E t$$

Ce qui prouve que E_1 est trivial et est ainsi conmtraire à l'hypothèse. Il est ainsi absurde de supposer que $x \notin Var(w)$ On en déduit alors que $x, y \in Var(w)$.□

Proposition 6.15 *Il n'existe pas de spécification E_1 de eq telle que $\forall u, v \in T_1$, $eq(u, v) =_E true \Leftrightarrow u =_E v$.*

Remarquons que, comme, lorsque $u, v \in T_1$, $u =_E v \Leftrightarrow \phi(u) =_{\mathbf{Z}} \phi(v)$, cette proposition signifie qu'on ne peut spécifier l'égalité sur les entiers.

Preuve

Nous allons classer les équations de la spécification: pour chaque équation $eq(u, v) == w$, on a $|Var(w)|, |Var(u, v)| \in \{0, 1, 2\}$ et ces deux ensembles de variables peuvent avoir diverses intersections. On pourra toujours supposer, par raison de symétrie, que $|Var(w)| \leq |Var(u, v)|$ (Si w n'est ni *true* ni *false*, w est de la forme $eq(u', v')$). D'autre part, comme $=_E$ est symétrique, si eq vérifie $eq(u, v) =_E true \Leftrightarrow u =_E v$, alors eq vérifie les hypothèses du lemme 6.14 et donc $|Var(u, v)| = 2 \Rightarrow Var(u, v) = Var(w)$. Les axiomes de E_1 sont donc (à symétrie près) de l'une des formes suivantes:

1. $eq(u, v) == eq(u', v')$ avec $Var(u) = Var(u') = \{x\}$ et $Var(v) = Var(v') = \{y\}$ et $x \not\equiv y$
2. $eq(u, v) == eq(u', v')$ avec $Var(u, v) = \{x\}$ et $Var(u', v') = \{y\}$ et $x \not\equiv y$
3. $eq(u, v) == eq(u', v')$ avec $Var(u) = Var(v) = Var(u') = Var(v') = \{x\}$
4. $eq(u, v) == eq(u', v')$ avec $Var(u) = Var(v) = Var(u') = \{x\}$ et $Var(v') = \emptyset$
5. $eq(u, v) == w$ avec $Var(u) = Var(v) = \{x\}$ et $Var(w) = \emptyset$
6. $eq(u, v) == eq(u', v')$ avec $Var(u) = Var(u') = \{x\}$ et $Var(v) = Var(v') = \emptyset$
7. $eq(u, v) == w$ avec $Var(u) = \{x\}$ et $Var(v) = Var(w) = \emptyset$
8. $eq(u, v) == w$ avec $Var(u, v, w) = \emptyset$

Informellement, le cas 1 est celui où l'un des deux membres possède deux variables distinctes. Les cas 2,3,4,6 sont les cas où les deux membres possèdent une variable et une seule. les cas 5,7 correspondent aux cas où l'un des deux membres possède une et une seule variable et l'autre aucune. Le dernier cas correspond au cas où aucun des deux membres ne possède de variable.

Nous allons éliminer ou préciser chacun de ces cas:

Dans le cas 1, $\phi(u) - \phi(v) = \phi(u') - \phi(v')$

d'après le lemme 6.13 on peut en effet trouver une substitution σ telle que $u\sigma =_E v\sigma$.

Par hypothèse sur eq, on a alors aussi $u'\sigma =_E v'\sigma$. Appliquant à nouveau le lemme 6.13, on obtient alors:

$$\phi(u) - \phi(v) + \phi(x\sigma) - \phi(y\sigma) = \phi(u') - \phi(v') + \phi(x\sigma) - \phi(y\sigma)$$

Ce qui fournit la propriété annoncée.

Dans le cas 2, nécéssairement $Var(u) = Var(v)$ et $Var(u') = Var(v')$

Supposons en effet que, par exemple, $Var(v') = \emptyset$. Alors, d'après le lemme 6.13, il existe des substitutions σ_1, σ_2 telles que $u'\sigma_1 =_E v'$ et $u'\sigma_2 \neq_E v'$. Comme $Var(u, v)$ ne rencontre pas le domaine de ces substitutions, on obtient: $eq(u, v) =_E eq(u'\sigma_1, v') =_E true$ et $eq(u, v) =_E eq(u'\sigma_2, v') =_E false$. Ce qui est bien entendu absurde.

Le cas 7 n'a pas lieu

Nous pouvons en effet faire le même raisonnement que ci-dessus: on peut trouver deux substitutions σ_1 et σ_2 telles que $u\sigma_1 =_E v$ et $u\sigma_2 \neq_E v$. On en déduit alors $true =_E false$.

Le cas 4 n'a pas lieu

En effet, ou bien $u =_E v$ et, en choisissant σ fermée de sorte que $u'\sigma \neq_E v'$, on trouve $true =_E false$, ou bien $u \neq_E v$ et, comme $Var(u) = Var(v)$, pour toute substitution σ, on a $u\sigma \neq_E v\sigma$ d'après le lemme 6.13. Mais alors, en choisissant σ de sorte que $u'\sigma =_E v'$, on obtient à nouveau une contradiction.

En résumé : si $eq(u, v)$ est l'un des membres d'une équation de E, ou bien $Var(u) = Var(v)$, ou bien l'autre membre est $eq(u', v')$ avec $\phi(u) - \phi(v) =_{\mathbf{Z}} \phi(u') - \phi(v')$.

Soit alors $N = 1 + \max(\{|\phi(u) - \phi(v)| \mid eq(u, v) = w \in E\})$. Soit $w_1 = eq(0, s^N(0))$.

Il n'existe aucun membre d'équation de E de la forme $eq(u, v)$ avec $var(u) = Var(v)$ qui filtre w_1. En effet, si $Var(u) = Var(v)$, pour toute substitution σ, $\phi(u\sigma) - \phi(v\sigma) = \phi(u) - \phi(v)$. Si donc $w_1 \vdash w_2$ (c'est à dire w_2 se déduit de w_1 en une étape de remplacement d'égaux par égaux), c'est par application d'une équation $eq(u, v) = eq(u', v')$ telle que $\phi(u) - \phi(v) = \phi(u') - \phi(v')$. Mais alors $w_2 \equiv eq(u_2, v_2)$ avec $\phi(u_2) - \phi(v_2) = -N$. Par suite, si $w_1 \vdash \ldots \vdash w_n$, alors $w_n \equiv eq(u_n, v_n)$ et $\phi(u_n, v_n) =_{\mathbf{Z}} -N$.

Ceci est en contradiction avec la propriété supposée de eq puisque $0 \neq_E s^N(0)$ et pourtant $eq(0, s^N(0)) \neq_E false$.

□

6.3.2 Comment résoudre le problème de la section précédente

Nous présentons succintement ici comment résoudre le problème ci-dessus. L'idée présentée ici a été développée (après la soutenance de thèse) dans [Com88a]. Nous invitons donc le lecteur à se reporter à cet article pour plus de détails.

Remarquons que, si, au lieu d'une spécification multi-sorte, nous considérons une spécification avec sortes ordonnées des entiers relatifs (cf chapitre 4 pour les définitions):

$$0 : \rightarrow \underline{zero} \qquad \begin{array}{ll} s : & \underline{pos} \rightarrow \underline{pos} \\ & \underline{zero} \rightarrow \underline{pos} \end{array} \qquad \begin{array}{ll} p : & \underline{neg} \rightarrow \underline{neg} \\ & \underline{zero} \rightarrow \underline{neg} \end{array} \qquad eq : \underline{int} \times \underline{int} \rightarrow \underline{bool}$$

avec l'ordre sur les sortes : $\underline{int} > \underline{pos}$, $\underline{int} > \underline{neg}$ et $\underline{int} > \underline{zero}$, alors il est facile de définir complètement eq par un système d'équations. En effet, il n'y a pas, dans cette spécification de "relation entre les constructeurs" $0, s, p$: l'algèbre des termes se compose de 0, des termes de la forme $s^n(0)$ et des termes de la forme $p^n(0)$. L'égalité peut donc se définir par:

$$\begin{aligned}
eq(x,y) &== eq(y,x) & \qquad eq(0,0) &== true\\
eq(0,x:\underline{pos}) &== false & eq(0,x:\underline{neg}) &== false\\
eq(s(x:\underline{pos}),s(y:\underline{pos})) &== eq(x,y) & eq(s(x:\underline{zero}),s(y:\underline{zero})) &== true\\
eq(p(x:\underline{neg}),p(y:\underline{neg})) &== eq(x,y) & eq(p(x:\underline{zero}),p(y:\underline{zero})) &== true\\
eq(s(x:\underline{zero}),s(y:\underline{pos})) &== false & eq(p(x:\underline{zero}),p(y:\underline{neg})) &== false\\
eq(x:\underline{pos},y:\underline{neg}) &== false
\end{aligned}$$

Cette propriété n'est pas surprenante. D'une part nous avons vu dans le chapitre 4 que les signatures avec sortes ordonnées ne sont autres que des automates d'arbres d'états finis. D'autre part, nous avons vu dans le chapitre 5 comment calculer une grammaire (régulière si possible) du langage des termes fermés irréductibles pour un système de réécriture $\mathcal{R}$. Mais ce langage (NF) est isomorphe à l'algèbre initiale $T(F)/=_E$ lorsque $\mathcal{R}$ est convergent. En rapprochant ces deux résultats, nous pouvons déduire une structure d'algèbre (libre) avec sortes ordonnées sur $T(F)/=_E$.

La construction de cette structure est décrite en détail dans [Com88a] où il est de plus prouvé que les théorèmes inductifs sont conservés (i.e. sont les mêmes que l'on considère $T(F)/=_E$ comme une F-algèbre ou comme une algèbre avec sortes ordonnées ...). Limitons nous ici à l'exemple et à une description partielle du calcul de la structure de sortes ordonnées.

Le système de réécriture $\mathcal{R} = \{s(p(x)) \to x;\ p(s(x)) \to x\}$ est convergent. Il est de plus aisé de calculer une grammaire du langage des formes normales (cf chapitre 5) :

$$\begin{array}{lcl}
NF & \to & NF_0 \mid NF_{s(x)} \mid NF_{p(x)}\\
NF_0 & \to & 0\\
NF_{s(x)} & \to & s(NF_0) \mid s(NF_{s(x)})\\
NF_{p(x)} & \to & p(NF_0) \mid p(NF_{p(x)})
\end{array}$$

A chaque non-terminal est associé une sorte (ou un état de l'automate). Les ϵ transitions (ou 1-règles) définissent les relations entre sortes. Ici, à NF est associée la sorte $\underline{int}$, à NF_0 la sorte $\underline{zero}$, à $NF_{s(x)}$ la sorte $\underline{pos}$ et à $NF_{p(x)}$ la sorte $\underline{neg}$. La première série de règles donne les relations $\underline{int} > \underline{pos}$, $\underline{int} > \underline{neg}$ et $\underline{int} > \underline{zero}$.

Cette construction se généralise à n'importe quelle grammaire régulière du langage des formes normales. Il faut, de façon générale, ajouter la déclaration d'inclusion de sorte $\underline{s} > \underline{s'}$ lorsque $\underline{s}$ est associé à NF_t, $\underline{s'}$ est associé à NF_u et u est une instance de t. C'est le cas ici si l'on remarque que NF n'est qu'une abréviation pour NF_x.

A chaque règle de grammaire qui n'est pas une 1-règle on associe alors un profil d'opérateur: si $NF_t \to f(NF_{t_1}, \ldots, NF_{t_n})$ on associe à f le profil $f : \underline{s_1} \times \ldots \times \underline{s_n} \to \underline{s}$ si $\underline{s_i}$ est la sorte associée à NF_{t_i} et $\underline{s}$ est la sorte associée à NF_t. Dans notre exemple, on associe à s les profils $s : \underline{zero} \to \underline{pos}$ et $s : \underline{pos} \to \underline{pos}$ à cause des règles de grammaire $NF_{s(x)} \to s(NF_0)$ et $NF_{s(x)} \to s(NF_{s(x)})$. Dans l'exemple des entiers, on obtient ainsi la signature avec sortes ordonnées:

$$0 :\rightarrow \underline{zero}$$
$$s : \underline{zero} \rightarrow \underline{pos} \qquad s : \underline{pos} \rightarrow \underline{pos}$$
$$\text{'}p : \underline{zero} \rightarrow \underline{neg} \qquad p : \underline{neg} \rightarrow \underline{neg}$$

Il faut remarquer que, d'une part si un symbole de fonction f était convertible à $F-\{f\}$, alors il n'a plus aucun profil dans cette nouvelle signature. D'autre part que certains symboles qui étaient définis partout peuvent n'être plus que des fonctions partielles. Par exemple s n'est plus défini sur les entiers négatifs dans cette nouvelle spécification. Nous avons donc seulement calculé une signature avec sortes ordonnées $(S', \geq, F')$ telle que:

Proposition 6.16 *NF est une $(S', \geq, F')$ algèbre libre.*

Preuves commentaires et détails se trouvent dans [Com88a].

L'étape suivante consiste à définir les "anciens" symboles de fonction dans cette nouvelle présentation de sorte à obtenir des algèbres isomorphes (voir [Com88a]).

Une conséquence de telles transformations est de ramener n'importe quelle spécification avec "relations entre constructeurs" à une présentation avec sortes ordonnées et constructeurs libres, à condition que les relations entre constructeurs (dans la spécification de départ) définissent un quotient reconnaissable (au sens des langages d'arbres).

Une telle transformation présente de nombreus intérêts. Nous avons montré le problème de la spécification de l'égalité sur les entiers. Dans [Com88a] est abordé le problème de l'automatisation des preuves par induction dans les théories équationnelles: une telle transformation *permet d'éviter les tests de réductibilité inductive* et d'utiliser l'algorithme de Huet et Hullot [HH82] au lieu de celui de Jouannaud et Kounalis [JK86b]. (Voir aussi l'introduction du chapitre 5 et [Com88a]).

Chapitre 7

E-disunification

7.1 Limites théoriques

L'unification dans les théories équationnelles est connue pour être un problème indécidable, mais il existe de nombreuses procédures complètes dont la "plus efficace" est sans doute (pour l'instant) celle de J. Gallier et W. Snyder [GS87].

Les problèmes équationnels contenant en particulier les problèmes d'unification, il n'est pas question de chercher à obtenir une procédure de décision de la satisfaisabilité dans le cas des théories équationnelles. Mais on peut se demander s'il existe une procédure permettant d'énumérer les solutions d'un problème équationnel dans une théorie équationnelle. Autrement dit; peut-on généraliser la méthode de Gallier et Snyder ?

La réponse est non car l'ensemble des solutions dans $T(F)/=_E$ d'un problème équationnel n'est pas même récursivement énumérable:

Théoreme 7.1 *Il n'existe pas de procédure P qui, étant donné un ensemble d'axiomes équationnels E et un problème équationnel $\mathcal{P}$ ayant une solution dans $T(F)/=_E$, calcule une solution $P(\mathcal{P}, E)$ en temps fini.*

Autrement dit, la résolution des problèmes équationnels dans les théories équationnelles n'est pas même semi-décidable.

Preuve

Supposons qu'une telle procédure existe et considérons le problème de la résolution des systèmes d'équations diophantiennes (10ème problème de Hilbert). Soit $\mathcal{P}$ un système d'équations diophantiennes: $\mathcal{P} \equiv u_1 = v_1 \wedge \ldots \wedge u_n = v_n$. Soit $\vec{x} = Var(\mathcal{P})$ et soit $\mathcal{P}'$ le problème:

$$\mathcal{P}' \equiv \forall \vec{x} : u_1 \neq v_1 \vee \ldots \vee u_n \neq v_n$$

d'inconnue principale $x_0 \notin \vec{x}$. $\mathcal{P}$ ou $\mathcal{P}'$ admet une solution sur les entiers. Donc $P(\mathcal{P}, E)$ ou bien $P(\mathcal{P}', E)$ termine. Construisant alors une procédure qui "exécute en parallèle" (i.e. effectue alternativement un pas d'éxécution de chacune des procédures) $P(\mathcal{P}, E)$ et $P(\mathcal{P}', E)$ on obtient une procédure qui termine toujours et permet de décider de l'existence d'une solution à un système d'équations diophantiennes. Ce qui est contradictoire avec la fameuse indécidabilité de ce problème. □

Ceci montre certaines limites aux résultats que l'on peut espérer obtenir sur la E-disunification. On peut d'ailleurs noter que ce n'est pas la présence de quantificateurs mais la négation qui apporte de telles limites:

Corollaire 7.2 *Il n'existe pas d'algorithme qui associe à chaque ensemble fini d'axiomes E et à chaque problème équationnel $\mathcal{P}$, soluble (dans $T(F)/=_E$) et sans paramètre une solution de $\mathcal{P}$ dans $T(F)/=_E$.*

Preuve
Il suffit de montrer que le problème de la résolution des problèmes équationnels avec paramètres dans une théorie E se ramène à la résolution des problèmes équationnels sans paramètre dans une théorie E'. Plus précisément même, vu la preuve donnée ci-dessus, il suffit de prouver que le problème $\forall Var(u,v) : u \neq v$ possède des solutions dans $T(F)/=_E$ si et seulement si $u' \neq v'$ possède des solutions dans $T(F')/=_{E'}$. Il suffit enfin pour cela d'ajouter à E les équations $u' = u$ et $v' = v$ où u' et v' sont des nouveaux symboles de constante. u' et v' ne sont pas dans la même classe modulo E' ssi il n'existe aucune substitution σ telle que $u\sigma$ et $v\sigma$ soient dans la même classe modulo E. □

Dans la suite, nous ne chercherons donc pas à généraliser des procédures de semi-décision de l'unification et nous ne considérerons que des théories équationnelles pour lesquelles l'unification est décidable. Notons que ce n'est pas suffisant pour assurer la décidabilité de la disunification puisqu'il existe[1] des théories équationnelles pour lesquelles l'unification est décidable et le problème du mot est indécidable. Or la résolution des problèmes équationnels contient en particulier le problème du mot.

7.2 Transformations dans les théories équationnelles

Les difficultés qui surgissent lorsque l'on veut généraliser les résultats du chapitre 3 aux théories équationnelles viennent du fait que certaines règles de transformation ne sont pas correctes (ou pas adéquates) lorsque $\mathcal{A}$ n'est pas une algèbre libre. Ces restrictions sont indiquées clairement dans l'énoncé de la proposition 3.5.

On peut envisager plusieurs façons d'aborder ces difficultés:

1. on peut introduire des règles de "mutation" ([Kir85,Kir86]) qui font intervenir les axiomes de la théorie et qui permettent de "récupérer" la complétude lorsque les règles ne sont pas adéquates. On peut considérer que la procédure de Gallier et Snyder [GS87] rentre dans cette catégorie.

2. on peut, plutôt que d'ajouter des règles à celles du chapitre 3, modifier celles qui ne sont pas adéquates en les "généralisant". Par exemple, la décomposition se généralise aux théories syntaxiques [Kir85].

3. on peut enfin adopter un point de vue totalement différent en essayant de ramener la disunification à l'unification. C'est à dire à essayer de généraliser la méthode de A. Colmerauer [Col84] aux théories équationnelles. C'est cette approche que suivent M.

[1] Communication privée de M. Schmidt-Schauss.

Maher [Mah88a] et HJ. Bürckert [Bur88]. Ce dernier envisage justement l'extension aux théories équationnelles. Mais il se restreint aux problèmes de complément et ne donne pas de résultat de solubilité dans $T(F)/=_E$.

Cette dernière approche pose de nombreux problèmes. Le premier d'entre eux est que, dans les théories équationnelles, le "principe d'indépendance des diséquations" [LMM86] n'est plus correct, même pour les problèmes sans paramètres. Or, comme il est montré dans [LMM86], ce principe est la clef de voûte des résultats de solubilité de A. Colmerauer [Col82]. Nous verrons quand même dans la section 4 qu'il est possible de généraliser ces résultats à certaines théories équationnelles.

Le deuxième problème posé par la dernière approche est celui de l'élimination des paramètres. (On peut noter qu'il n'est pas abordé dans [Bur88]). Les règles d'élimination des paramètres des chapitres 3 et 4 (ainsi que celles ue nous donnerons par la suite) supposent toujours que le paramètre est membre d'une équation ou d'une diséquation. La règle d'explosion avait en effet pour but de "faire remonter" les paramètres afin d'appliquer une des règles d'élimination. Le principe était simple : si le paramètre y apparait à une position interne de t dans $z \neq t$, on "explose" z, puis, par décomposition, la taille de la psoition de y décroit. Une telle méthode n'est envisageable dans les théories équationnelles que si l'on dispose d'une règle pouvant se substituter à la règle de décomposition. L'utilisation directe d'un algorithme d'unification ne permet pas de contrôler la taille des positions des paramètres.

Pour conclure, nous excluons la troisième méthode, au moins lorsque les problèmes équationnels peuvent contenir des paramètres.

Il reste à discuter les mérites respectifs des deux premières approches. Notons que, dans le premier cas, si l'on peut bien récupérer des résultats de complétude en ajoutant des règles, il faut quand même que toutes les règles du système initial soient correctes. Or, par négation, s'il y a des règles inadéquates, il y en a qui ne sont pas correctes. Cette technique de "récupération de la complétude" par ajout de règles est donc mal adaptée à des problèmes contenant des négations.

Nous allons donc tenter dans cette section de généraliser au maximum les règles qui ne sont pas correctes ou pas adéquates dans les théories équationnelles soient:

- les règles de décomposition
- les règles d'incompatibilité
- les tests d'ocurrence
- l'élmination des paramètres dans les équations

Les autres règles du chapitre 3 sont conservées ici.

Les idées développées s'appuient largement sur celles de C. Kirchner [Kir85].

7.2.1 Décompositions et incompatibilités des équations

Rappelons que les équations de E sont supposées non orientées et sont notées avec le signe $==$. Si bien que $u == v$ est identique à $v == u$.

Comme dans [Kir85] les décompositions se généralisent bien aux théories "syntaxiques":

Définition 7.3 *Un ensemble d'axiomes (équationnels) E est* syntaxique *si, pour tous termes $t_1, \ldots, t_n, u_1, \ldots, u_p$ dont les variables ne rencontrent pas les variables des équations de E, $f(t_1, \ldots, t_n) =_E g(u_1, \ldots, u_p)$ si et seulement si*

- *ou bien $f = g$ et, pour tout i, $t_i =_E u_i$*
- *ou bien il existe une équation $f(v_1, \ldots, v_n) == g(w_1, \ldots, w_p)$ dans E et une substitution σ telles que, pour tout indice i, $v_i\sigma =_E t_i$ et $w_i\sigma =_E u_i$.*

Une théorie est syntaxique *si elle peut être engendrée par un ensemble fini E d'axiomes syntaxiques.*

Remarque:
Cette définition est un tout petit peu plus générale que celle de C. Kirchner puisqu'elle autorise les axiomes potents.

Exemple 7.1 Les théories composées d'axiomes de commutativité sont syntaxiques [Kir85]. La théorie (EA) composée des axiomes d'élément absorbant à gauche et à droite : $y*0 == 0$ et $0 * y == 0$ est syntaxique.

La théorie (A) composée de l'axiome d'associativité $x + (y + z) == (x + y) + z$ est syntaxique.

En fait, pour énoncer nos règles, nous pouvons ne considérer qu'une classe plus générale de théories équationnelles:

Définition 7.4 *Un ensemble d'axiomes (équationnels) E est* presque syntaxique *si, pour tous termes $t_1, \ldots, t_n, u_1, \ldots, u_p$ dont les variables ne rencontrent pas celles des équations de E, $f(t_1, \ldots, t_n) =_E g(u_1, \ldots, u_p)$ si et seulement si:*

- *ou bien $f = g$ et, pour tout i, $t_i =_E u_i$*
- *ou bien il existe une équation $f(v_1, \ldots, v_n) == g(w_1, \ldots, w_p) \in E$ et une substitution σ tels que, pour tous i, j, $v_i\sigma =_E t_i$ et $w_j\sigma =_E u_j\sigma$.*
- *ou bien il existe une équation $f(v_1, \ldots, v_n) == w \in E$ (ou $g(v_1, \ldots, v_p) == w \in E$), il existe un indice i et une substitution σ tels que w soit un sous-terme de v_i et, pour tout indice j, $v_j\sigma =_E t_j$ et $w\sigma =_E g(u_1, \ldots u_p)$.*

Une théorie est presque syntaxique *si elle peut être engendrée par un ensemble fini d'axiomes presque syntaxique.*

Une propriété des théories syntaxiques (ou presque syntaxiques) est que, dans chaque preuve de $u =_E v$, il y a au plus une inférence appliquée à la position ϵ. (Voir [Kir85] pour plus de précisions). Ceci donne une idée intuitive de ce que sont ces théories. Donnons en de toutes façons quelques exemples typiques.

Exemple 7.2 Bien sûr, les théories syntaxiques sont presque syntaxiques. Les théories de l'exemple 7.1 sont donc des théories presque syntaxiques. Mais il est possible que des ensembles d'axiomes soient presque syntaxiques sans être syntaxiques, comme les deux ensembles :

$$\begin{array}{lllllllll} EN: & 0+y & == & y & & & & & \\ CEN': & y+y' & == & y'+y & y+0 & == & y & 0+y & == & y \end{array}$$

Lemme 7.5 *Les ensembles d'axiomes EN et CEN' sont presque syntaxiques mais pas syntaxiques*[2].

Preuve

Montrons tout d'abord que EN est une théorie presque syntaxique. Il suffit de prouver que $(u'+v')\sigma =_{EN} (u+v)\sigma$ entraine

$$\begin{array}{ll} & (u'\sigma =_{EN} u\sigma \text{ et } v'\sigma =_{EN} v\sigma) \\ \text{ou} & (u\sigma =_{EN} 0 \text{ et } v\sigma =_{EN} u'\sigma + v'\sigma) \\ \text{ou} & (u'\sigma =_{EN} 0 \text{ et } v'\sigma =_{EN} u\sigma + v\sigma) \end{array}$$

On note plus simplement $u_1 \equiv u\sigma$, $u_2 \equiv u'\sigma$, $v_1 \equiv v\sigma$ et $v_2 \equiv v'\sigma$. EN peut être orientée en la règle $0+y \rightarrow y$ qui constitue un système de réécriture canonique. Donc $u_1 + v_1 =_{EN} u_2 + v_2$ si et seulement si leurs formes irréductibles pour ce système de réécriture, soient $(u_1+v_1)\downarrow$ et $(u_2+v_2)\downarrow$, sont syntaxiquement égales. Quatre cas se présentent alors:

1. $u_1\downarrow \not\equiv 0$ et $u_2\downarrow \not\equiv 0$. Dans ce cas $(u_1+v_1)\downarrow \equiv u_1\downarrow + v_1\downarrow$ et $(u_2+v_2)\downarrow \equiv u_2\downarrow + v_2\downarrow$ d'où l'on déduit $u_1\downarrow \equiv u_2\downarrow$ et $v_1\downarrow \equiv v_2\downarrow$. C'est à dire $u_1 =_{EN} u_2$ et $v_1 =_{EN} v_2$.
2. $u_1\downarrow \not\equiv 0$ et $u_2\downarrow \equiv 0$. Dans ce cas, $(u_1+v_1)\downarrow \equiv u_1\downarrow + v_1\downarrow$ et $(u_2+v_2)\downarrow \equiv v_2\downarrow$. D'où $v_2 =_{EN} u_1 + v_1$ et $u_2 =_{EN} 0$
3. $u_1\downarrow \equiv 0$ et $u_2\downarrow \not\equiv 0$. C'est le cas symétrique du précédent. On obtient $v_1 =_{EN} u_2 + v_2$ et $u_1 =_{EN} 0$.
4. $u_1\downarrow \equiv 0$ et $u_2\downarrow \equiv 0$. Dans ce cas, $(u_1+v_1)\downarrow \equiv v_1\downarrow$ et $(u_2+v_2)\downarrow \equiv v_2\downarrow$. D'où $u_1 =_{EN} u_2 =_{EN} 0$ et $v_1 =_{EN} v_2$

Dans chacun des cas on obtient le résultat voulu.

EN et CEN' ne sont pas syntaxiques de façon évidente: $0+(x_1+x_2) =_E x_1+x_2$ mais $x_1 \neq_E 0$ et $x_2 \neq_E 0$.

Il ne reste plus qu'à prouver que CEN' est presque syntaxique. La preuve s'effectue comme ci-dessus pour EN en considérant la réécriture modulo la commutativité. Nous laissons les détails au lecteur. □

[2]Par contre, il est probable (c'est une conjecture que nous n'avons pas étudiée pour l'instant) qu'il y ait identité entre les théories syntaxiques et les théories presque syntaxiques. Il semblerait qu'il suffise d'ajouter certaines conséquences équationnelles à un ensemble d'axiomes presque syntaxique pour obtenir un ensemble syntaxique. Par exemple, dans le cas de EN, si l'on ajoute léquation $0+(x_1+x_2) == x_1+x_2$, on obtient un ensemble d'axiomes syntaxique.

(ED_1) $\mathcal{P}[f(t_1,\ldots,t_n) = f(u_1,\ldots,u_n)] \mapsto \exists Var(E) : \mathcal{P}[$

$$
\begin{array}{ll}
 & (t_1 = u_1 \wedge \ldots \wedge t_n = u_n) \\
\vee & \bigvee_{f(v_1,\ldots,v_n)==f(w_1,\ldots,w_n)\in E}(t_1 = v_1 \wedge \ldots \wedge t_n = v_n \wedge u_1 = w_1 \wedge \ldots \wedge u_n = w_n) \\
\vee & \bigvee_{f(v_1,\ldots,v_n)==w\in E_1}(t_1 = v_1 \wedge \ldots \wedge t_n = v_n \wedge f(u_1,\ldots,u_n) = w) \\
\vee & \bigvee_{f(v_1,\ldots,v_n)==w\in E_1}(u_1 = v_1 \wedge \ldots \wedge u_n = v_n \wedge f(t_1,\ldots,t_n) = w) \\
 &]
\end{array}
$$

(EI_1) $\mathcal{P}[f(t_1,\ldots,t_n) = g(u_1,\ldots,u_p)] \mapsto \exists Var(E) : \mathcal{P}[$

$$
\begin{array}{ll}
 & \bigvee_{f(v_1,\ldots,v_n)==g(w_1,\ldots,w_p)\in E}(t_1 = v_1 \wedge \ldots \wedge t_n = v_n \wedge u_1 = w_1 \wedge \ldots \wedge u_p = w_p) \\
\vee & \bigvee_{f(v_1,\ldots,v_n)==w\in E_1}(t_1 = v_1 \wedge \ldots \wedge t_n = v_n \wedge g(u_1,\ldots,u_p) = w) \\
\vee & \bigvee_{g(v_1,\ldots,v_p)==w\in E_1}(u_1 = v_1 \wedge \ldots \wedge u_p = v_p \wedge f(t_1,\ldots,t_n) = w) \\
 &]
\end{array}
$$

Dans ces règles, $f \neq g$ et E_1 est le sous-ensemble de E formé des équations de la forme $f(v_1,\ldots,v_n) == w$ où w est un sous-terme de l'un des v_i. On suppose de plus les équations de E renommées de sorte que $Var(E) \cap Var(\mathcal{P}) = \emptyset$.
Noter aussi que la disjonction indexée par un ensemble vide est, par convention, $\perp$.

Figure 7.1: Décompositions dans les théories presque syntaxiques

Les règles de la figure 7.1 généralisent les règles de décomposition du chapitre 3 (et celles de C. Kirchner). Notons que les problèmes ne sont pas en forme normale conjonctive. Nous avons en effet préféré ne pas alourdir en évitant de développer.

Proposition 7.6 *Lorsque E est presque syntaxique, les règles ED_1 et EI_1 de la figure 7.1 sont fortement adéquates par rapport à $\mathcal{A} = T(F,X)/=_E$ (et $\mathcal{A} = T(F)/=_E$) et à $\mathcal{I}$ tel que $Var(E) \cap \mathcal{I} = \emptyset$.*

Preuve
La correction des règles est immédiates (elle est d'ailleurs indépendante de la théorie considérée) et la forte adéquation est une conséquence directe de la définition d'un ensemble d'axiomes presque syntaxique. □

la proposition suivante établit un premier résultat de terminaison tout en montrant les limites de la méthode.

Proposition 7.7 *Soit E un ensemble d'axiomes presque syntaxique tel que,*

1. *pour toute équation $u == v \in E$, u et v sont deux termes de profondeur inférieure ou égale à 2*
2. *si $u == v \in E$ et u est un sous-terme strict de v alors v est de profondeur 1*

Alors l'application des règles ED_1 et EI_1 termine.

Preuve

Afin de simplifier, nous supposerons que les problèmes sont en forme normale disjonctive.

Soit $\phi_2(s = t)$ le couple (a, b) des profondeurs des deux termes s et t, ordonné de sorte que $a < b$. ϕ_2 est défini de même sur les diséquations.

Soit ϕ_1 la fonction définie sur les conjonctions d'équations et de diséquations par: $\phi_1(e_1 \wedge \ldots \wedge e_n)$ est le multi-ensemble des couples $\phi_2(e_i)$.

Soit ϕ la fonction qui à tout problème équationnel en forme normale disjonctive associe le multi-ensemble:

$$\phi(\exists \vec{w}, \forall \vec{y} : c_1 \vee \ldots \vee c_n) = \{\phi_1(c_1), \ldots, \phi 1(c_n)\}$$

Montrons que ϕ est décroissante (strictement) par application de ED_1 ou de EI_1.

Soit $(a, b) = \phi_2(f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n))$ et supposons que $\mathcal{P} \mapsto_{ED_1} \mathcal{P}'$. (Le cas de EI_1 est analogue). On suppose de plus les deux problèmes en forme normale disjonctive. Notons $(a_i, b_i) = \phi_2(t_i = u_i)$. Si

$$\phi(\mathcal{P}) = \{M_1, \ldots, M_k, \{c_1, \ldots, c_m, (a, b)\}\}$$

alors

$$\begin{aligned}\phi(\mathcal{P}') = \{ \; & M_1, \ldots, m_k, \{c_1, \ldots, c_m, (a_1, b_1), \ldots, (a_n, b_n)\}, \\ & \{c_1, \ldots, c_m, c'_{1,1}, \ldots, c'_{1,n}, c''_{1,1}, \ldots, c''_{1,n}\}, \ldots, \{c_1, \ldots, c_m, c'_{j,1}, \ldots, c'_{j,n}, c''_{j,1}, \ldots, c''_{j,n}\}, \\ & \{c_1, \ldots, c_m, d_{1,1}, \ldots, d_{1,n}, (0, a)\}, \ldots, \{c_1, \ldots, c_m, d_{r,1}, \ldots, d_{r,n}, (0, a)\}, \\ & \{c_1, \ldots, c_m, d'_{1,1}, \ldots, d'_{1,n}, (0, b)\}, \ldots, \{c_1, \ldots, c_m, d'_{r,1}, \ldots, d'_{r,n}, (0, b)\} \\ \} & \end{aligned}$$

Chacune des trois dernières lignes correspondant aux trois grandes disjonctions dans la règle: $c'_{i,p}$ est le couple $\phi_2(v_p, t_p)$ pour la $i$ème équation de E de la forme $f(v_1, \ldots, v_n) == f(w_1, \ldots, w_n)$ et, de même, $c''_{i,p}$ est $\phi_2(w_p, u_p)$. Les couples $d_{i,p}$, $d'_{i,p}$ étant construits de façon analogue. Remarquons alors que

- pour tout i, $(a_i, b_i) < (a, b)$
- pour tous i, p, $c'_{i,p} < (a, b)$ et $c''_{i,p} < (a, b)$ car $a \geq 1$ et $c'_{i,p}$, $c''_{i,p}$ sont de l'une des formes suivantes : $(0, a_i)$, $(1, a_i)$, $(0, b_i)$, $(1, b_i)$. Et, si $a = 1$, a_i et b_i sont strictement inférieurs à b.
- Pour tous i, p, $d_{i,p} < (a, b)$ et $d'_{i,p} < (a, b)$ par le même raisonnement

D'où la décroissance de ϕ. □

Les hypothèses de la proposition 7.7 sont bien toutes nécéssaires (hélas !) en effet:

Exemple 7.3

Soit $E = \{f(f(f(x_1, x_2), x_3), x_4) == f(x_1, x_4)\}$. Montrons que la décomposition ne termine pas[3] et donc qu'on ne peut élargir le résultat de la proposition 7.7 à des ensembles d'axiomes contenant des termes de profondeur 3:

$$f(f(z_1, z_2), z_3) = f(z_4, z_5) \mapsto_{ED_1} \mathcal{P}[f(z_1, z_2) = f(f(x_1, x_2), x_3)]$$

Cette deuxième équation étant, à renommage près, l'équation de départ.

[3]Nous n'avons pas prouvé que E est syntaxique, mais cet exemple me semble suffisant pour illustrer la nécéssité des hypothèses.

(ED_2) $P \wedge (d \vee f(t_1,\ldots,t_n)) \mapsto \forall Var(E) : P \wedge (d \vee ($

$$
\begin{array}{ll}
 & (t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n) \\
\wedge & \bigwedge_{f(v_1,\ldots,v_n)==f(w_1,\ldots,w_n)\in E}(t_1 \neq v_1 \vee \ldots \vee t_n \neq v_n \vee u_1 \neq w_1 \vee \ldots \vee u_n \neq w_n) \\
\wedge & \bigwedge_{f(v_1,\ldots,v_n)==w\in E_1}(t_1 \neq v_1 \vee \ldots \vee t_n \neq v_n \vee f(t_1,\ldots,t_n) \neq w) \\
\wedge & \bigwedge_{f(v_1,\ldots,v_n)==w\in E_1}(u_1 \neq v_1 \vee \ldots \vee u_n \neq v_n \vee f(t_1,\ldots,t_n) \neq w) \\
)) &
\end{array}
$$

(EI_2) $P \wedge (d \vee f(t_1,\ldots,t_n) \neq g(u_1,\ldots,u_p)) \mapsto \forall Var(E) : P \wedge (d \vee ($

$$
\begin{array}{ll}
 & \bigwedge_{f(v_1,\ldots,v_n)==g(w_1,\ldots,w_p)\in E}(t_1 \neq v_1 \vee \ldots \vee t_n \neq v_n \vee u_1 \neq w_1 \vee \ldots \vee u_n \neq w_n) \\
\wedge & \bigwedge_{f(v_1,\ldots,v_n)==w\in E_1}(t_1 \neq v_1 \vee \ldots \vee t_n \neq v_n \vee g(u_1,\ldots,u_p) \neq w) \\
\wedge & \bigwedge_{g(v_1,\ldots,v_p)==w\in E_1}(u_1 \neq v_1 \vee \ldots \vee u_p \neq v_p \vee f(t_1,\ldots,t_n) \neq w) \\
)) &
\end{array}
$$

Dans ces règles, $f \neq g$ et E_1 est l'ensemble des équations de E qui sont de la forme $u == w$ avec w sous-terme strict de u.

Figure 7.2: Décomposition et incompatibilité des diséquations dans les théories presque syntaxiques

Exemple 7.4 Soit $E = \{f(f(a,x),a) == a\}$.

$$f(a,x) = a \mapsto_{EI_1} \exists w : a = f(a,w) \wedge x = a \wedge a = a$$

Par conséquent, l'application de la règle EI_1 seule ne termine pas lorsqu'appliquée à $f(a,x) = a$: la deuxième condition dans l'énoncé de la proposition 7.7 est nécessaire.

7.2.2 Décomposition et incompatibilité des diséquations

Proposition 7.8 *Les règles ED_2 et EI_2 sont fortement adéquates pour $\mathcal{A} = T(F)/=_E$ (ou $\mathcal{A} = T(F,X)/=_E$).*

Preuve

Il nous suffit ici de prendre la négation des règles de la figure 7.1 pour obtenir les règles de décomposition et d'incompatibilité de diséquations de la figure 7.2. La proposition 7.6 entraine alors la correction et la forte adéquation des règles ED_2 et EI_2 lorsque E est presque syntaxique.□

Remarques

- Une conjonction sur un ensemble vide étant, par convention, égale à $\top$, on retrouve les règles du chapitre 3 lorsque $E = \emptyset$.
- Comme les paramètres ajoutés sont toujours $Var(E)$, il faut utiliser à chaque fois une variante renommée de E.

Exemple 7.5 Considérons le problème $\forall y : x + 0 \neq y + x'$ dans la théorie CEN'. (qui est presque syntaxique).

$$
\begin{array}{lll}
\forall y : x + 0 \neq y + x' & \mapsto_{ED_2} & \forall y, y_1, y_2 : (x \neq y \vee 0 \neq x') \\
& & \wedge (x \neq y_1 \vee 0 \neq y_2 \vee y \neq y_2 \vee x' \neq y_1) \\
& & \wedge (x \neq y_2 \vee 0 \neq y_1 \vee y \neq y_1 \vee x' \neq y_2) \\
& & \wedge (x \neq 0 \vee 0 \neq y_1 \vee y_1 \neq y + x') \\
& & \wedge (0 \neq y \vee y_1 \neq x' \vee y_1 \neq x + 0) \\
& & \wedge (x \neq y_1 \vee 0 \neq 0 \vee y_1 \neq y + x') \\
& & \wedge (y \neq y_1 \vee 0 \neq x' \vee y_1 \neq x + 0) \\
& \mapsto^*_{EP_i} & \forall y : x' \neq 0 \wedge x \neq x' \wedge (x \neq 0 \vee y + x' \neq 0) \\
& & \wedge x' \neq x + 0 \wedge x \neq y + x' \\
& \mapsto_{EI_2} & \forall y, y' : x' \neq 0 \wedge x \neq x' \wedge (x \neq 0 \vee (\\
& & \quad (y \neq 0 \vee y' \neq x' \vee y' \neq 0) \wedge (y' \neq y \vee x' \neq 0 \vee y' \neq 0)) \\
& & \wedge x' \neq x + 0 \wedge x \neq y + x' \\
& \mapsto^*_{EP_i} & \forall y : x' \neq 0 \wedge x \neq x' \wedge (x \neq 0 \vee x' \neq 0) \\
& & \wedge x' \neq x + 0 \wedge x \neq y + x'
\end{array}
$$

Les règles EP_i sont les règles d'élimination des paramètres dont la validité ne dépend pas du modèle (c'est à dire les règles EP_1 et EP_2 de la figure 3.1).

Le dernier problème obtenu ci-dessus peut être considéré comme totalement décomposé. Malheureusement, il subsiste des occurrences de paramètres. Dans la théorie vide, nous en viendrions à bout en utilisant l'explosion de x. Hélas, l'application d'une telle règle sur cet exemple conduit à la non-terminaison.

L'exemple illustre donc bien le problème auquel nous allons être confronté quand nous voudrons obtenir des résultats de terminaison: les règles de décomposition et d'incompatibilité de diséquations introduisent de nouveaux paramètres et provoquent des "boucles" avec les règles d'élimination de paramètres.

7.2.3 Tests d'occurrence

La façon la plus simple de traiter les tests d'occurrence est de se limiter aux théories dans lesquels ils restent corrects (théories *strictes* dans [Kir85]). Il est néanmoins possible d'obtenir un peu mieux sans effort. Nous généralisons donc un petit peu ici la notion de théorie stricte en admettant des axiomes "potents".

Définition 7.9 *Un ensemble fini E d'axiomes équationnels est* presque strict *si, pour tous termes $t_1, \ldots, t_n$ et tout $x \in Var(t_1, \ldots, t_n)$ tels que $Var(E) \cap Var(t_1, \ldots, t_n) = \emptyset$, $f(t_1, \ldots, t_n) =_E x$ si et seulement si il existe une équation $f(u_1, \ldots, u_n) == w \in E$ et une substitution σ telles que:*

- *w est un sous-terme de l'un des u_i*
- *pour tout indice j, $t_j =_E u_j\sigma$ et $w\sigma =_E x$*

Une théorie est presque stricte *s'il existe un ensemble d'axiomes presque strict qui l'engendre.*

Les ensembles d'axiomes C, AC, EN, CEN',... sont des exemples d'ensembles presque stricts.

(EO_1) $\mathcal{P}[f(t_1,\ldots,t_n) = x] \mapsto \exists Var(E) : \mathcal{P}[\bigvee_{f(u_1,\ldots,u_n)==w\in E_1} (t_1 = u_1 \wedge \ldots \wedge t_n = u_n \wedge x = w)]$

(EO_2) $\mathcal{P}[f(t_1,\ldots,t_n \neq x] \mapsto \forall Var(E) : \mathcal{P}[\bigwedge_{f(u_1,\ldots,u_n)==w\in E_1} (t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n \vee x \neq w)]$

Si E_1 est les sous-ensemble de E formé des axiomes de la forme $f(u_1,\ldots,u_n) == w$ où w est un sous-terme de l'un des v_i, x est une variable de l'un des t_i et $Var(E)$ est disjoint des variables de $\mathcal{P}$ et de $\mathcal{I}$.

Figure 7.3: Tests d'occurrence dans les théories presque strictes

En conservant les conventions sur les conjonctions et les disjonctions indicées par des ensembles vides, on obtient les règles de généralisation des tests d'occurrence de la figure 7.3.

Proposition 7.10 *Les règles EO_1 et EO_2 sont fortement adéquates par rapport à $\mathcal{A} = T(F,X)/=_E$ (ou $\mathcal{A} = T(F)/=_E$) lorsque E est presque stricte.*

7.2.4 Élimination des paramètres des équations

Nous poursuivons ici la généralisation des règles de la figure 3.2. D'après la proposition 3.5, la règle EP_4 est fortement adéquate dans tout quotient de $T(F)$ (ou de $T(F,X)$). Il ne reste donc que les règles EP_3 et MF_4 (parmi les règles de la figure 3.2) dont nous n'avons pas encore étudié l'extension aux théories équationnelles. C'est ce que nous faisons dans ce paragraphe.

Proposition 7.11 *EEP et EMF (avec le contrôle de la figure 7.4) sont fortement adéquates pour $\mathcal{A} = T(F)/=_E$.*

La preuve est immédiate.

Les règles EEP et EMF de la figure 7.4 sont des restrictions des règles EP_3 et MF_4 parce que l'on exige (pour EEP) que z_i soit un paramètre (au lieu d'exiger seulement que $z_i = u_i$ contienne une occurrence de paramètre). On exige aussi que $\{y_1,\ldots,y_n\} \cap Var(t_1,\ldots,t_n) = \emptyset$ alors que EP_3 exigeait seulement $y_i \not\equiv t_i$. Ces restrictions peuvent en fait être affaiblies dans le cas des théories compactes (cf section 4) mais certaines restrictions sont nécessaires en général car EP_3 n'est pas correcte dans le cas des théories équationnelles. Par exemple, $\forall y : y = y + 0$ est valide dans EN et ne doit donc pas être réduite à $\perp$ par application de EP_3.

7.3 Disunification dans les théories quasi-libres

Comme dans le chapitre 3, l'idée pour résoudre les problèmes équationnels (et plus généralement les formules équationnelles) est de commencer par éliminer les paramètres. Malheureuse-

$$(EEP)\ \forall \vec{y} : P \wedge (y_1 = t_1 \vee \ldots \vee y_n = t_n \vee d) \mapsto \forall \vec{y} : P \wedge d$$

Si

1. $y_1, \ldots, y_n \in \vec{y}$
2. d est une disjonction d'équations et de diséquations qui ne contient pas de paramètre
3. $Var(t_1, \ldots, t_n) \cap \{y_1, \ldots, y_n\} = \emptyset$
4. Pour tout i, $T(F)/=_E$ contient une infinité de classes de même sorte que y_i

$$(EMF)\ \exists \vec{w} : P \wedge (d_1 \vee w \neq t_1) \wedge \ldots \wedge (d_n \vee w \neq t_n) \mapsto \exists \vec{w} : P$$

Si $w \in \vec{w}$ et $w \notin Var(P) \cup Var(t_1, \ldots, t_n)$ et il y a une infinité de classes dans $T(F)/=_E$ de même sorte que celle de w.

Figure 7.4: Règle d'élimination des paramètres dans les théories équationnelles

ment, nous avons vu dans la section précédente qu'il est possible de "boucler" en enchainant élimination de paramètres et décompositions de diséquations. Or ces décompositions sont nécessaires lorsque l'un des membres de la diséquation contient une occurrence (interne) de paramètre. De façon générale il y a donc un problème de terminaison. C'est pourquoi nous nous limitons ici au cas des théories *quasi-libres* qui sont des théories dans lesquelles les paramètres introduits par les règles ED_2 et EI_2 sont immédiatement éliminés.

Définition 7.12 *Un ensemble fini d'axiomes équationnels E est* quasi-libre *s'il est syntaxique, strict et si toute équation de E est de la forme $u == v$ où u et v sont de profondeur inférieure ou égale à 1. Une théorie équationnelle est* quasi-libre *si elle peut être engendrée par un ensemble d'axiomes quasi-libre.*

Il y a peu d'exemples de théories quasi-libres. On peut citer les combinaisons d'axiomes de commutativité (ou plus généralement d'axiomes permutatifs. La classe des théories quasi-libres contient la classe des théories permutatives de Mal'cev [Mal71]). On peut aussi citer EA (définie précédemment). Mais il faut reconnaitre que cette classe n'est pas très large (d'où le nom de "quasi-libre").

Les règles de la figure 7.5 sont obtenues par combinaison des règles ED_2, EI_2 et EP_2 qui permet de ne pas introduire de nouveau paramètre, lorsque l'ensemble d'axiomes est quasi-libre.

Proposition 7.13 *Les règles QL_1 et QL_2 sont fortement adéquates dans $T(F,X)/=_E$ (resp. $T(F)/=_E$) lorsque E est quasi-libre.*

(QL_1) $f(t_1,\ldots,t_n) \neq f(u_1,\ldots,u_n) \mapsto$

$$(t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n)$$
$$\bigwedge_{f(v_1,\ldots,v_n)==f(w_1,\ldots,w_n)\in E}\Big(\big(\bigvee_{v_i\equiv v_j \text{ et } v_i \text{ variable}} t_i \neq t_j\big) \vee \big(\bigvee_{v_i \text{ constante}} t_i \neq v_i\big) \vee \big(\bigvee_{w_i\equiv v_j \text{ et } v_j \text{ variable}} u_i \neq t_j\big) \vee \big(\bigvee_{w_i \text{ constante}} u_i \neq w_i\big) \vee \big(\bigvee_{w_i\equiv w_j \text{ et } w_j \text{ variable}} u_i \neq u_j\big)\Big)$$

(QL_2) $f(t_1,\ldots,t_n) \neq g(u_1,\ldots,u_p) \mapsto$

$$\bigwedge_{f(v_1,\ldots,v_n)==g(w_1,\ldots,w_p)\in E}\Big(\big(\bigvee_{v_i\equiv v_j \text{ et } v_i \text{ variable}} t_i \neq t_j\big) \vee \big(\bigvee_{v_i \text{ constante}} t_i \neq v_i\big) \vee \big(\bigvee_{w_i\equiv v_j \text{ et } v_j \text{ variable}} u_i \neq t_j\big) \vee \big(\bigvee_{w_i \text{ constante}} u_i \neq w_i\big) \vee \big(\bigvee_{w_i\equiv w_j \text{ et } w_j \text{ variable}} u_i \neq u_j\big)\Big)$$

Si $f \neq g$

Figure 7.5: Règles de transformation particulières aux théories quasi-libres

Preuve
Les règles QL_1 et QL_2 ne sont que des combinaisons des règles EI_2, ED_2 respectivement et de la ègle EP_2 du chapitre 3. Cette proposition est donc une conséquence des propositions 3.4 et 7.8. □

En combinant toutes les règles obtenues, nous obtenons alors un ensemble de règles dont l'application non déterministe termine et fournit des systèmes en forme résolue, généralisant ainsi le théorème 3.16.

L'ensemble des règles utilisées ici est récapitulé dans les figures 7.6, 7.7 et 7.8.

Théoreme 7.14 *Les règles des figures 7.6, 7.7,7.8 sont toutes correctes et adéquates dans $\mathcal{A} = T(F)/=_E$ où E est un ensemble d'axiomes quasi-libre. Leur application non déterministe termine. Les problèmes équationnels irréductibles pour ces règles ne contiennent pas de paramètre.*

Preuve
La correction et l'adéquation des règles utilisées est une conséquence des propositions 3.4, 3.5, 3.8, 7.6, 7.8, 7.13. Sauf en ce qui concerne les règles EI'_1 et ED'_1 qui tirent parti du fait que les théories quasi-libres sont supposées syntaxiques (et pas seulement quasi-syntaxiques). Mais la forte adéquation de ces règles découle immédiatement de la définition 7.3. Notons aussi que la forte adéquation des tests d'occurrence vient du fait que les théories quasi-libres sont supposées strictes.

Elimination des paramètres (EP)

$$(EP_1) \quad \forall \vec{y}, y : P \mapsto \forall \vec{y} : P \qquad \text{Si } y \notin Var(P)$$

$$(EP_2) \quad \forall \vec{y} : P \wedge (y \neq t \vee d) \mapsto \forall \vec{y} : P \wedge d\{y \to t\} \qquad \text{Si } d \text{ est une disjonction d'équations et de diséquations, } y \in \vec{y} \text{ et } y \notin Var(t)$$

$$(EEP) \quad \forall \vec{y} : P \wedge (y_1 = t_1 \vee \ldots \vee y_n = t_n \vee d) \mapsto \forall \vec{y} : P \wedge d$$

Si

1. $y_1, \ldots, y_n \in \vec{y}$
2. d est une disjonction d'équations et de diséquations qui ne contient pas de paramètre
3. $Var(t_1, \ldots, t_n) \cap \{y_1, \ldots, y_n\} = \emptyset$
4. Pour tout i, $T(F)/=_E$ contient une infinité de classes de même sorte que y_i

$$(EP_4) \quad \forall \vec{y} : P \wedge Q \mapsto \forall \vec{y} : P \wedge Q\{y \to t_1\} \wedge \ldots \wedge Q\{y \to t_n\}$$

Si y est un paramètre de sorte $\underline{s}$ dont le support dans $T(F)/=_E$ est $\{\overline{t_1}, \ldots, \overline{t_n}\}$.

Incompatibilités (I)

$$(EI_1') \quad \mathcal{P}[f(t_1, \ldots, t_n) = g(u_1, \ldots, u_p)] \mapsto \exists Var(E) : \mathcal{P}[\bigvee_{f(v_1,\ldots,v_n)==g(w_1,\ldots,w_p)\in E} (t_1 = v_1 \wedge \ldots \wedge t_n = v_n \wedge u_1 = w_1 \wedge \ldots \wedge u_p = w_p)]$$

Si $f \neq g$

$$(QL_2) \quad f(t_1, \ldots, t_n) \neq g(u_1, \ldots, u_p) \mapsto$$

$$\bigwedge_{f(v_1,\ldots,v_n)==g(w_1,\ldots,w_p)\in E} \Big(\big(\bigvee_{v_i \equiv v_j \text{ et } v_i \text{ variable}} t_i \neq t_j\big) \vee \big(\bigvee_{v_i \text{ constante}} t_i \neq v_i\big) \vee \big(\bigvee_{w_i \equiv v_j \text{ et } v_j \text{ variable}} u_i \neq t_j\big) \vee \big(\bigvee_{w_i \text{ constante}} u_i \neq w_i\big) \vee \big(\bigvee_{w_i \equiv w_j \text{ et } w_j \text{ variable}} u_i \neq u_j\big)\Big)$$

Si $f \neq g$

Pour les règles d'incompatibilité, on supposera que l'un des membres de l'équation ou de la diséquation contient une occurrence de paramètre. Noter aussi que la disjonction indexée par un ensemble vide est, par convention $\bot$.

Figure 7.6: Élimination des paramètres dans les théories quasi-libres: partie I

Elimination des équations et diséquations triviales (T)

$$\begin{array}{lll} (T_1) & s = s & \mapsto \top \\ (T_2) & s \neq s & \mapsto \bot \end{array}$$

Tests d'occurrence (O)

$$\begin{array}{lll} (O_1) & z = t \mapsto \bot & \text{Si } z \in Var(t) \\ (O_2) & z \neq t \mapsto \top & \text{Si } z \in Var(t) \end{array}$$

Fusions (F)

$$\begin{array}{ll} (F_1) & z = t \wedge z = u \mapsto z = t \wedge t = u \\ (F_3) & z = t \wedge z \neq u \mapsto z = t \wedge t \neq u \\ (F_1') & z = t \wedge (z = u \vee d) \mapsto z = t \wedge (t = u \vee d) \\ (F_3') & z = t \wedge (z \neq u \vee d) \mapsto z = t \wedge (t \neq u \vee d) \end{array}$$

Pour ces règles de fusion, on supposera que:

1. z est une inconnue et pas t
2. t ne contient pas d'occurrence de paramètre
3. u contient une occurrence de paramètre et n'est pas lui-même un paramètre

$$\begin{array}{ll} (F_2) & z \neq t \vee z \neq u \mapsto z \neq t \vee t \neq u \\ (F_4) & z = u \vee z \neq t \mapsto u = t \vee z \neq t \end{array}$$

Pour ces règles de fusion, on supposera que:

1. z est une variable et t n'est pas une variable
2. u contient une occurrence de paramètre
3. Ou bien taille-param(t) $\leq$ taille-param(u) ou bien u est un paramètre résolu.

Figure 7.7: Élimination des paramètres dans les théories quasi-libres: partie II

Décompositions (D)

$$(ED'_1)\ \mathcal{P}[f(t_1,\ldots,t_n) = f(u_1,\ldots,u_n)] \mapsto \exists Var(E) : \mathcal{P}[(t_1 = u_1 \wedge \ldots \wedge t_n = u_n) \vee \bigvee_{f(v_1,\ldots,v_n)==f(w_1,\ldots,w_n)\in E}(t_1 = v_1 \wedge \ldots \wedge t_n = v_n \wedge u_1 = w_1 \wedge \ldots \wedge u_n = w_n)]$$

$$(QL_1)\ f(t_1,\ldots,t_n) \neq f(u_1,\ldots,u_n) \mapsto$$

$$(t_1 \neq u_1 \vee \ldots \vee t_n \neq u_n) \wedge_{f(v_1,\ldots,v_n)==f(w_1,\ldots,w_n)\in E}((\bigvee_{v_i\equiv v_j \text{ et } v_i \text{ variable}} t_i \neq t_j) \vee (\bigvee_{v_i \text{ constante}} t_i \neq v_i) \vee (\bigvee_{w_i\equiv v_j \text{ et } v_j \text{ variable}} u_i \neq t_j) \vee (\bigvee_{w_i \text{ constante}} u_i \neq w_i) \vee (\bigvee_{w_i\equiv w_j \text{ et } w_j \text{ variable}} u_i \neq u_j))$$

Pour les règles de décomposition, on supposera que $f(t_1,\ldots,t_n)$ ou $f(u_1,\ldots,u_n)$ contient au moins une occurrence de paramètre.

Explosion (E)

$$(Ex_1)\ \forall \vec{y} : P \mapsto \exists w_1,\ldots,w_p, \forall \vec{y} : P \wedge x = f(w_1,\ldots,w_p)$$

Cette règle ne sera appliquée que si

1. x est une inconnue et $\vec{w} \cap (Var(P) \cup \vec{y} \cup \mathcal{I}) = \emptyset$ et $f \in F$
2. Il existe une équation $x = u$ (ou une diséquation $x \neq u$) dans P telle que u n'est pas une variable et contient au moins une occurrence de paramètre.
3. Aucune des règles **EP**, **F**, **D**, **I**, **O**, **T** ne peut s'appliquer.

Figure 7.8: Élimination des paramètres dans les théories quasi-libres: partie III

La preuve de terminaison est quant à elle identique à celle du théorème 3.16. (Nous avons en fait tout fait pour obtenir cela !). Il suffit en fait de montrer que les fonctions d'interprétation définies dans la preuve du théorème 3.16 décroissent pour les nouvelles règles introduites ici (i.e. $EEP, ED'_1, EI'_1, QL_1, QL_2$). Rappelons en les définitions:

- Si d est une disjonction d'équations et de diséquations, $\phi_1(d)$ désigne le nombre de paramètres distincts ayant au moins une occurrence dans d.
- Etant donné une disjonction d'équations et de diséquations $d \equiv e_1 \vee \ldots \vee e_n$, $\phi_2(d)$ désigne le multi-ensemble $\{TM(e_1), \ldots, TM(e_n)\}$ où $TM(e)$ est défini par:
  - $TM(e) = 0$ si l'un des membres de e est un paramètre résolu
  - Sinon, $TM(s = t) = TM(s \neq t) =$ max(taille-param(s),taille-param(t)).
- Si, à nouveau, d est une disjonction d'équations et de diséquations, $\phi_3(d)$ est le nombre d'équations et de diséquations de d dont un des membres est une variable.
- Si $\mathcal{P} \equiv \exists \vec{w}, \forall \vec{y} : d_1 \wedge \ldots \wedge d_n$ est un problème en forme normale conjonctive, $\psi_1(\mathcal{P})$ est le multi-ensemble de triplets

$$\{(\phi_1(d_1), \phi_2(d_1), \phi_3(d_1)), \ldots, (\phi_1(d_n), \phi_2(d_n), \phi_3(d_n))\}$$

- Si $\mathcal{P}$ est un problème équationnel en forme normale conjonctive, $\psi_2(\mathcal{P})$ est la taille totale de $\mathcal{P}$, c'est-à-dire le nombre total de symboles de $F \cup X$ apparaissant dans $\mathcal{P}$.
- $\Phi(\mathcal{P})$ est le couple $(\psi_1(\mathcal{P}), \psi_2(\mathcal{P}))$

Vérifions maintenant la décroissance de Φ par application des (nouvelles) règles:

- EEP fait décroitre ϕ_1, donc ψ_1
- Supposons que $\mathcal{P} \mapsto_{ED'_1} \mathcal{P}$ où $\mathcal{P}$ et $\mathcal{P}'$ sont en forme normale conjonctive et montrons que $\Phi(\mathcal{P}) > \Phi(\mathcal{P}')$. Soit

$$\psi_1(\mathcal{P}) = \{d_1, \ldots, d_n, (a, \{, b_1, \ldots, b_k, TM(f(t_1, \ldots, t_m) = f(u_1, \ldots, u_m))\}, c)\}$$

Soient $E_1, \ldots, E_N$ les équations de E de la forme:

$$E_i \equiv f(v_{i,1}, \ldots, v_{i,m}) == f(w_{i,1}, \ldots, w_{i,m})$$

On note Γ l'ensemble des applications e qui associent à $j \in \{1, \ldots, N\}$ une des équations $t_j = v_{j,1}, \ldots, t_m = v_{j,m}, u_1 = w_{j,1}, \ldots, u_m = w_{j,m}$. Alors $\psi_1(\mathcal{P}')$ peut s'écrire:

$$\psi_1(\mathcal{P}') = \{d_1, \ldots, d_n\} + \sum_{1 \leq i \leq m} \sum_{e \in \Gamma} \{(a_{i,e}\{b_1, \ldots, b_k, TM(e(1)), \ldots, TM(e(N))\}, c_{i,e})\}$$

Si bien que $\psi_1(\mathcal{P}')$ est un multi-ensemble de $n + m * m^N$ triplets. Cette "explosion combinatoire" correspondant à la mise en forme normale conjonctive: chaque triplet correspond à une disjonction de $\mathcal{P}'$; à i et $e \in \Gamma$ correspond la disjonction

$t_i = u_i \vee e(1) \vee \ldots \vee e(l)$. La première composante $a_{i,e}$ de chaque triplet est le nombre de paramètres apparaissant dans la disjonction. Comme aucun paramètre n'a été introduit, pour chaque i, e, $a_{i,e} \leq a$. De plus, par hypothèse (contrôle), $TM(f(t_1, \ldots, t_m) = f(u_1, \ldots, u_m)) \geq 1$ et, par conséquent, pour tout $e \in \Gamma$ et tout $1 \leq l \leq N$, $TM(e(l)) < TM(f(t_1, \ldots, t_m) = f(u_1, \ldots, u_m))$. D'où la décroissance de ψ_1.

- De la même façon, ψ_1 est strictement décroissante par application de EI'_1.
- Supposons maintenant que $\mathcal{P} \mapsto_{QL_1} \mathcal{P}'$. Soit, comme ci-dessus,

$$\psi_1(\mathcal{P}) = \{d_1, \ldots, d_n, (a, \{b_1, \ldots, b_k, TM(f(t_1, \ldots, t_m) = f(u_1, \ldots, u_m))\}, c)\}$$

Alors, $\psi_1(\mathcal{P}')$ peut s'écrire :

$$\begin{aligned}\psi_1(\mathcal{P}') = \{d_1, \ldots, d_n, (a, \{b_1, \ldots, b_k, TM(t_1 \neq u_1), \ldots, TM(t_m \neq u_m)\}, c')\} \\ + \textstyle\sum_{e \equiv f(v_1, \ldots, v_n) == f(w_1, \ldots, w_n) \in E} \{(a_e, \{b_1, \ldots, b_k\} \\ + \textstyle\sum_{v_i \equiv v_j \text{ et } v_i \text{ variable}} \{TM(t_i \neq t_j)\} \\ + \textstyle\sum_{v_i \text{ constante}} \{TM(t_i \neq v_i)\} \\ + \textstyle\sum_{w_i \equiv v_j \text{ et } v_j \text{ variable}} \{TM(u_i \neq t_j)\} \\ + \textstyle\sum_{w_i \text{ constante}} \{TM(u_i \neq w_i)\} \\ + \textstyle\sum_{w_i \equiv w_j \text{ et } w_i \text{ variable}} \{TM(u_i \neq u_j)\} \\ , c_e)\}\end{aligned}$$

Il suffit alors de remarquer que, pour tout e, $a_e \leq a$ et, pour tous i, j,

$$TM(t_i \neq t_j), TM(u_i \neq t_j), TM(u_i \neq u_j) < TM(f(t_1, \ldots, t_m) = f(u_1, \ldots, u_m))$$

Par hypothèse sur le contrôle. De plus $TM(t_i \neq v_i)$, lorsque v_i est une constante, est la somme des tailles des positions des paramètres dans t_i et est donc bien inféreure à $TM(f(t_1, \ldots, t_m) = f(u_1, \ldots, u_m))$. (De même pour $TM(u_i \neq w_i)$).

D'où le résultat de décroissance de ψ_1 par application de QL_1.

- De même, ψ_1 décroit strictement par application de (QL_2)

Le reste de la preuve du théorème est rigoureusement la même que celle du théorème 3.16 et n'est donc pas reproduite ici. □

Le théorème 7.14 établit un résultat de complétude vis à vis de l'élimination des paramètres, mais ce sont en fait tous les résultats concernant les problèmes équationnels étudiés dans les chapitres 3 et 4 qui s'étendent aux théories quasi-libres, comme nous allons le voir dans la section suivante.

7.4 Disunification dans les théories compactes

L'objectif est de donner un algorithme de décision de la validité de formules équationnelles dans $T(F)/=_E$. Nous essayons ici d'élargir la classe des théories considérées (nous ne voulons plus considérer seulement les théories quasi-libres, mais une classe qui contienne

au moins les théories AC). En contrepartie, il nous faut nous restreindre sur le type de formules équationnelles considérées.

En fait, nous étudions ici les théories "compactes" qui constituent un cas particulier de théories finitaires (au sens de [BHS87] par exemple) mais qui contiennent strictement les théories quasi-libres. Nous verrons plus loin que les théories AC sont compactes. Nous n'envisagerons aussi que les formules équationnelles purement existentielles.

Ces conditions peuvent paraitre très restrictives. Présentons donc les résultats de cette section sous un autre angle : nous généralisons ici les résultats de [Col84] au cas de certaines théories équationnelles, en particulier les théories AC.

Rappelons que nous notons $\mathcal{P} \approx_{\mathcal{A},\mathcal{I}} \mathcal{P}'$ lorsque les problèmes équationnels $\mathcal{P}$ et $\mathcal{P}'$ vérifient $\mathcal{S}(\mathcal{A},\mathcal{P},\mathcal{I}) = \mathcal{S}(\mathcal{A},\mathcal{P}',\mathcal{I})$. Nous emploierons dans ce paragraphe la notation abrégée $\mathcal{P} \approx_E \mathcal{P}'$ pour $\mathcal{P} \approx_{T(F)/=_E,\mathcal{I}} \mathcal{P}'$. La relation $\approx_E$ est étendue comme précédemment aux ensembles de problèmes équationnels (cette relation est alors compatible avec les opérations sur les ensembles).

Rappelons aussi qu'un *problème d'unification* est un problème équationnel sans négation et sans quantificateur universel. Nous dirons ici qu'un problème d'unification est *complètement résolu* s'il est de la forme

$$\exists \vec{w} : x_1 = t_1 \wedge \ldots \wedge x_n = t_n$$

avec $\mathcal{I} = \{x_1, \ldots, x_n\}$ et $\mathcal{I} \cap Var(\vec{w}, t_1, \ldots, t_n) = \emptyset$. Autrement dit, un problème d'unification complètement résolu désigne une unique substitution idempotente. On peut remarquer que les formes résolues (pour le système $\mathcal{R}_2$ du chapitre 3) de problèmes d'unification sont des problèmes complètement résolus.

Définition 7.15 *Un ensemble fini d'axiomes équationnels E est dit* finitaire *s'il existe un algorithme permettant de transformer tout problème d'unification $\mathcal{P}$ en un ensemble fini de problèmes complètement résolus $\mathcal{E}$ tel que $\{\mathcal{P}\} \approx_E \mathcal{E}$.*

Dans les théories finitaires (i.e. définies par un ensemble d'axiomes finitaire), on dispose donc d'une règle de "simplification des équations":

$$(ES)\, t_1 = u_1 \wedge \ldots \wedge t_n = u_n \mapsto Q \quad \text{Si } Q \in \mathcal{E} \text{ et } \mathcal{E} \approx_E t_1 = u_1 \wedge \ldots \wedge t_n = u_n$$

$\mathcal{E}$ étant bien entendu un ensemble de problèmes complètement résolus.

Cette règle, toujours correcte, est adéquate dans les théories finitaires. Elle est fortment adéquates dans les théories unitaires.

De même, il est possible, dans les théories finitaires, de décider de la validité d'une équation dans $T(F)/=_E$. On peut donc remplacer les règles d'élimination des équations triviales et des diséquations triviales par les rèvgles:

$$\begin{array}{lllll} (ET_1) & t = s & \mapsto & \top & \text{Si } t =_E s \\ (ET_2) & t \neq s & \mapsto & \bot & \text{Si } t =_E s \end{array}$$

La définition qui suit s'inspire de [LMM86]: des diséquations sont indépendantes si, lorsque chacune d'elles possède une solution, alors leur conjonction possède une solution.

Définition 7.16 *Les diséquations* $t_1 \neq u_1, \ldots, t_n \neq u_n$ *sont dites* indépendantes *(par rapport à l'ensemble d'axiomes E) si*

- *ou bien il existe un indice i tel que* $t_i =_E u_i$
- *ou bien une variable de* $Var(t_1, u_1, \ldots, t_n, u_n)$ *est d'une sorte* $\underline{s}$ *à support fini dans* $T(F)/=_E$
- *ou bien* $t_1 \neq u_1 \wedge \ldots \wedge t_n \neq u_n$ *a au moins une solution dans* $T(F)/=_E$.

Lorsque des diséquations sont indépendantes, pour résoudre leur conjonction, il suffit de résoudre chacune d'entre elles.

Définition 7.17 *Un ensemble E d'axiomes est dit* compact *s'il est finitaire et si n diséquations quelconques sont indépendantes par rapport à E.*

Remarquons que les théories libres (i.e. $E = \emptyset$) sont compactes d'après le lemme 3.7.

Nous nous intéressons aux théories compactes car l'application de la règle ED_2 est inutile dans ces théories. Or nous avons vu dans la section 2 de ce chapitre que c'est essentiellement cette règle qui pose problème dans les théories équationnelles parce qu'elle ajoute des paramètres au problème. En plus, s'il est inutile de décomposer les diséquations, il est aussi inutile de décomposer les équations lorsque l'on peut appliquer la règle ES. S'il n'y a pas lieu de décomposer, rien n'oblige alors à se limiter aux théories presque syntaxiques. De fait, nous verrons que les théories AC sont compactes.

Cela montre aussi pourquoi nous allons nous limiter aux problèmes purement existentiels : lorsqu'il y a des paramètres, pour les éliminer il nous faut d'abord les "faire remonter" et pour cela, une décomposition est nécessaire.

Par contre, il serait possible de généraliser tous les résultats de cette section aux théories où, au lieu d'être certain de l'existence d'une solution fermée à un système de diséquations, on dispose seulement d'un algorithme de décision de l'existence d'une telle solution.

7.4.1 Formes solubles dans les théories compactes

Partant de problèmes sans paramètres, nous utiliserons les règles suivantes : ES, ET_1, ET_2, R_1, Nc et la règle suivante qui permet d'éliminer les variables à support fini dans $T(F)/=_E$:

$$(Ex_2)\ \ P \mapsto x = t \wedge P\{x \rightarrow t\}$$

Si x est une variable non résolue de P dont la sorte
a un support fini dans $T(F)/=_E$ et t est l'un des représentants de cette sorte

Le remplacement de x par t (règle (R_1)) ne sera appliqué que si t ne contient aucune occurrence de x et x apparait dans au moins une diséquation.

D'autre part, on appliquera la règle (ES) seulement si le problème est SED (c'est à dire Sans equations dans les disjonctions, suivant la définition donnée dans le chapitre 4), elle sera appliquée à l'ensemble des équations du problème et seulement si cet ensemble d'équations n'est pas déjà complètement résolu.

Proposition 7.18 *L'application non déterministe des règles ci-dessus termine.*

Preuve
Remarquons tout d'abord que, sans la règle (ES) l'application des règles termine: aucune règle autre que (ES) ne permet l'introduction de nouvelles variables, le nombre de variables non résolues (cf chapitre 3 pour la définition) est donc décroissant. Seules les règles Nc, ET_1, ET_2 ne font pas nécessairement strictement décroitre cette fonction d'interprétation. Mais Nc elles font toutes trois strictement décroitre le nombre d'équations et de diséquations du problème.

Si maintenant $\mathcal{P}'$ est une forme irréductible de $\mathcal{P}$ pour toutes les règles sauf (ES), ou bien (ES) ne s'applique pas et nous avons terminé. Ou bien (ES) s'applique et le problème obtenu est de la forme

$$\mathcal{P}_1 \equiv \exists \vec{w} : x_1 = t_1 \wedge \ldots \wedge x_n = t_n \wedge d_1 \wedge \ldots \wedge d_m$$

où $d_1, \ldots, d_m$ sont des disjonctions d'une ou plusieurs diséquations. Toutes les transformations préservent cette forme de problème (à cause du contrôle). Par conséquent, la règle (ES) ne peut plus s'appliquer. Si $\mathcal{P}_1 \mapsto^* \mathcal{P}_2$, le nombre de variables de $\mathcal{P}_2$ est inférieur à celui de $\mathcal{P}_1$. On recommence alors le même raisonnement que ci-dessus: le nombre de variables non résolues décroit et, pour les règles pour lesquelles ce nombre ne décroit pas strictement, le nombre d'équations et de diséquations décroit strictement. □

Proposition 7.19 *Soit E un ensemble d'axiomes compact. Les problèmes équationnels sans paramètre irréductibles pour les règles ci-dessus possèdent au moins une solution dans $T(F)/=_E$.*

Preuve
Les problème irréductibles sont de la forme

$$\exists \vec{w} : x_1 = t_1 \wedge \ldots \wedge x_n = t_n \wedge u_1 \neq v_1 \wedge \ldots \wedge u_m \neq v_m$$

Où $x_1 = t_1 \wedge \ldots \wedge x_n = t_n$ est complètement résolu (puisque (ES) ne s'applique pas), les variables x_i n'ont qu'une occurrence dans le problème (puisque R_1 ne s'applique pas), $v_j \neq_E u_j$ (puisque ET_2 ne s'applique pas) et aucune variable de $Var(u_1, \ldots, u_m, v_1, \ldots, v_m)$ n'est de sorte à support fini dans $T(F)/=_E$ (puisque (Ex_2) ne s'applique pas). D'après les définitions 7.17 et 7.18, le système $u_1 \neq v_1 \wedge \ldots \wedge u_m \neq v_m$ possède donc une solution σ dans $T(F)/=_E$. $\{x_1 \rightarrow t_1\sigma; \ldots x_n \rightarrow t_n\sigma\}\sigma$ est alors une solution du problème dans $T(F)/=_E$. □

Corollaire 7.20 *La satisfaisabilité dans $T(F)/=_E$ d'une formule sans paramètre est décidable lorsque E est compacte.*

Par conséquent la validité d'une formule équationnelle dans $T(F)/=_E$ est décidable lorsque E est quasi-libre. (C'est le théorème 7.15).

7.4.2 Exemples de théories compactes

Proposition 7.21 *Les théories quasi-libres sont compactes.*

Preuve
Soit $t_1 \neq u_1 \wedge \ldots \wedge t_n \neq u_n$ un système de diséquations tel que, pour tout i, $t_i \neq_E u_i$ (E est un ensemble d'axiomes quasi-libre). Remarquons que les théories quasi-libres vérifient les hypothèses de la proposition 7.7. On peut donc se ramener au cas où, pour tout i, t_i est une variable. Il suffit alors de raisonner comme pour la preuve du lemme 3.7. □

Ce qui permet d'obtenir le résultat annoncé:

Théoreme 7.22 *Si E est un ensemble d'axiomes quasi-libre, $T(F)/=_E$ est complètement axiomatisable: la validité d'une formule équationnelle dans $T(F)/=_E$ est décidable.*

Ce résultat s'obtient comme dans le chapitre 3 en éléminant successivement tous les quantificateurs jusqu'à obtenir une formule purement existentielle (théorème 7.14). La décision de la validité d'une formule existentielle dans une théorie quasi-libre n'étant qu'un cas particulier du paragraphe précédent.

Définition 7.23 *Un ensemble d'équations E est AC s'il existe un sous-ensemble F' de F constitué de symboles binaires tel que E soit l'ensemble des équations $\{f(x,y) == f(y,x) \mid f \in F'\} \cup \{f(f(x,y),z) == f(x,f(y,z)) \mid f \in F'\}$.*

Proposition 7.24 *Tout ensemble d'équations AC est compact.*

Preuve
Soit $t_1 \neq u_1 \wedge \ldots \wedge t_n \neq u_n$ un système de diséquations tel que, pour tout i, $t_i \neq_E u_i$ et tel que toute variable du système soit à support infini dans $T(F)/=_E$. Il nous faut prouver que ce système possède au moins une solution fermée. Comme pour la preuve du lemme 3.7, raisonnons par récurrence sur le nombre de variables du système. S'il n'en a aucune, la proposition est triviale. Sinon, soit x une variable du sytème et montrons que $t_i = u_i$ n'a qu'un nombre fini de solutions si l'on considère toutes les variables distinctes de x comme des constantes[4].

Prouvons ce résultat par récurrence sur la profondeur minimale de t_i et de u_i:

- si l'un des deux termes t_i, u_i est de profondeur 0, alors (par exemple) t_i est x ou une constante. Si c'est une constante, u_i ne peut être cette même constante puisque $t_i \neq_E u_i$. Donc -ou bien $u_i \equiv x$ et l'on a l'unique solution $\{x \rightarrow t_i\}$ -ou bien u_i n'est pas une variable et il n'y a aucune solution.

[4]Ce résultat n'est pas une conséquence directe du fait que AC est finitaire. On peut en effet en déduire que $t_i = u_i$ possède un nombre fini de formes complètement résolues, mais, a priori, ces formes résolues peuvent contenir de "nouvelles" variables (i.e. qui ne sont ni dans t_i ni dans u_i. Et une équation $x = t[x']$ possède une infinité de solution fermées. Il nous faut donc analyser en détail l'unification AC dans ce cas particulier d'une unique variable pour montrer qu'il existe un ensemble de formes résolues équivalent qui possède pour seule variable x (les autres variables de t_i, u_i étant, rappelons le, considérées comme des constantes).

Si maintenant $t_i \equiv x$, u_i ne peut être x car $u_i \neq_E t_i$. Donc -ou bien u_i contient une occurrence de x et il n'y a pas de solution -ou bien u_i ne contient pas d'occurrence de x et $t_i = u_i$ possède pour unique solution $\{x \rightarrow u_i\}$.

Ainsi, lorsque l'un des deux termes est de profondeur 0, l'équation $t_i = u_i$ possède au plus une solution.

- Supposons que t_i et u_i sont de profondeur au moins 1. Trois cas se présentent :
  - Ou bien ils ont des symboles de tête distincts, et l'équation n'a pas de solution
  - Ou bien ils ont même symbole de tête et ce symbole n'est pas associatif-commutatif : $t_i = u_i \equiv f(t_{i,1}, \ldots, t_{i,n_1}) = f(u_{i,1}, \ldots, u_{i,n_1})$ et l'ensemble des solutions de $t_i = u_i$ est l'intersection des ensembles de solutions des équations $t_{i,j} = u_{i,j}$, $1 \leq j \leq n_1$ qui sont par hypothèse de récurrence tous finis. L'ensemble des solutions de $t_i = u_i$ est donc alors fini.
  - Ou bien ils ont le même symbole de tête : $+$ qui est associatif commutatif. Classiquement, on peut alors "mettre à plat" les deux termes, $+$ étant utilisé en notation infixée et sans parenthésage:

$$t_i = u_i \equiv n * x + v_1 + \ldots + v_{n_1} = n' * x + w_1 + \ldots + w_{n_2}$$

où $n * x$ désigne $\underbrace{x + \ldots + x}_{n}$ et $v_1, \ldots, v_{n_1}, w_1, \ldots, w_{n_2}$ sont des termes n'ayant pas $+$ pour symbole de tête. On peut supposer aussi $n \geq n'$ sans perdre de généralité. L'équation s'écrit alors

$$(n - n') * x + v_1 + \ldots + v_{n_1} = w_1 + \ldots + w_{n_2}$$

Si $n \neq n'$, alors cette équation possède au plus $n_2 - n_1$ solutions pour x,qui est nécessairement égal à l'un des w_i.
Si $n = n'$, alors $n_2 = n_1$ (sinon il n'y a pas de solution) et l'ensemble des solutions fermées de l'équation est contenu dans la réunion des solutions des équations $v_i = w_{\pi(i)}$ pour toute permutation π des indices. L'ensemble des solutions de $t_i = u_i$ est donc contenue dans une réunion finie d'ensembles qui sont finis par hypothèse de récurrence. D'où le résultat.

Ainsi, chaque équation $t_i = u_i$ a au plus un nombre fini de solutions lorsque les variables autres que x sont considérées comme des constantes.
On note $X_0 = Var(t_1, \ldots, t_n, u_1, \ldots, u_n) - \{x\}$. Soit C_x un sous-ensembre fini de $T(F, X_0)$ tel que σ soit solution dans $T(F, X_0)/ =_E$ de l'une des équations $t_i = u_i$ ssi il existe dans C_x un terme u tel que $\sigma = \{x \rightarrow \overline{u}\}$ ($\overline{u}$ désigne comme précédemment la classe d'équivalence de u modulo E). Par hypothèse x est de sorte à support infini dans $T(F)/ =_E$. Il existe donc un terme fermé $t \in T(F)$ qui n'est égal modulo E à aucun terme de C_x. Le système

$$t_1\{x \rightarrow t\} \neq u_1\{x \rightarrow t\} \wedge \ldots \wedge t_n\{x \rightarrow t\} \neq u_n\{x \rightarrow t\}$$

vérifie alors $\forall i, t_i\{x \rightarrow t\} \neq_E u_i\{x \rightarrow t\}$, ne contient pas de variable à support fini dans $T(F)/ =_E$ et possède une variable de moins que le système précédent (x n'apparait plus). Il est donc possible d'appliquer l'hypothèse de récurrence : ce système possède une solution θ dans $T(F)/ =_E$. Alors $\{x \rightarrow \overline{t}\}\theta$ est solution dans $T(F)/ =_E$ de $t_1 \neq u_1 \wedge \ldots \wedge t_n \neq u_n$. □

7.5 Conclusion

Combinant les résultats des deux sections précédentes, nous obtenons la décidabilité des formules équationnelles dans $T(F)/=_E$ lorsque E est quasi-libre (c'est le théorème 7.15). En d'autres termes toute algèbre de la classe des théories quasi-libres est complètement axiomatisable. Tous les résultats et toutes les applications envisagées ici (chapitres 5 et 6) s'étendent donc sans doute aux théories quasi-libres.

Néanmoins, ces résultats ne sont pas complètement satisfaisants puisque nous n'avons pas apporté de réponse à la question de la décision des formules équationnelles dans les théories compactes, et en particulier dans les théories AC. Les problèmes de la section précédente sont en effet purement existentiels. La disunification AC reste ainsi un problème ouvert : nous n'avons pas même de conjecture à émettre quant au résultat. On peut seulement remarquer que le problème risque d'être complexe: lorsque F se réduit à $\{a, +\}$, où $+$ est un symbole AC et a est une constante, $T(F)/=_E$ est complètement axiomatisable: il s'agit d'un codage de l'arithmétique de Presburger. Lorsque F contient un nombre fini de constantes et un symbole AC, le problème est celui de l'axiomatisation des multi-ensembles finis sur un ensemble fini. Il semble (?) pouvoir se ramener à l'arithmétique de Presburger.

Ces constatations indiquent une direction de recherche: ou bien tenter de coder le problème posé dans l'arithmétique de Presburger (s'il est décidable !) ou bien tenter de coder l'arithmétique de Péano (s'il est indécidable). Mais aucune de ces deux directions n'a pour l'instant été envisagée sérieusement. Il faut par contre noter l'échec des méthodes que nous avons employées tout au long de cette thèse pour résoudre le problème: en cas d'associativité-commutativité, il n'est pas possible d'effectuer des "transformations locales". Cette idée intuitive est reflétée par le fait que les théories AC ne sont pas syntaxiques. On remarque d'ailleurs que les algorithmes d'unification AC ([Sti81,Fag87]) sont des algorithmes "ad hoc" qui ramènent le problème à des résolutions de systèmes d'équations sur les entiers, et ne sont pas des instances de shémas plus généraux (cf [Kir85]).

Même en cas d'indécidabilité de la disunification AC, il faudrait se poser le problème de la résolution des problèmes de compléments dans une théorie AC puisque ce sont ces problèmes qui interviennent dans la plupart des applications.

Chapitre 8

Équations, diséquations et inéquations

La méthode de Jouannaud et Kounalis [JK86b] pour l'automatisation des preuves par induction dans les théories équationnelles, méthode que nous avons rappelée dans le chapitre 5, présente plusieurs inconvénients. L'un d'eux est que, utilisant la complétion des sytèmes de réécriture, il est possible de rencontrer un "cas d'échec", c'est à dire une équation non orientable.

J. Hsiang et M. Rusinowitch ont par ailleurs proposé une méthode de complétion sans échec [HR87] qui évite les cas d'échec et qui est ainsi réfutationnellement complète (pour le problème du mot). L'idée de base de la complétion sans échec est de considérer une équation comme orientée dans un sens ou dans l'autre, selon le terme auquel elle s'applique. Plus précisément si $s = t$ est une équation, $\geq$ un ordre de simplification total sur les termes fermés et u un terme, u *se réduit* en v par $s == t$ à la position p si

- ou bien il existe une substitution σ telle que $u/p \equiv s\sigma$, $v \equiv u[t\sigma]_p$ *et* $t\sigma \not\geq s\sigma$
- ou bien il existe une substitution σ telle que $u/p \equiv t\sigma$, $v \equiv u[s\sigma]_p$ *et* $s\sigma \not\geq t\sigma$

Si $(\mathcal{R}, E)$ est constitué d'un ensemble de règles de réécriture et d'un ensemble d'équations, s *se réduit* en t dans ce système, ce que l'on note $s \rightarrow_{\mathcal{R},E} t$ si $s \rightarrow_{\mathcal{R}} t$ ou bien il existe une position p de s et une équation $g == d$ de E telles que s se réduise en t par $g == d$ à la position p.

Suivant cette idée de complétion sans échec, L. Bachmair [Bac88] a étendu la méthode de Jouannaud et Kounalis pour automatiser les preuves par induction dans les théories équationnelles tout en évitant les cas d'échec. Il obtient ainsi un système réfutationnellement complet pour les preuves par induction.

Néanmoins L. Bachmair suppose que l'on peut tester la réductibilité inductive d'une équation par le système $(\mathcal{R}_0, E_0)$ de départ. Or il est possible que ce système contienne une équation non orientable (par exemple la commutativité $x + y == y + x$), i.e. que E_0 soit nécessairement non vide. Dans ce cas, la notion de réduction est modifiée, comme nous l'avons vu ci-dessus. Donc la notion de réductibilité inductive aussi. Et il n'existe pas (à l'heure actuelle) de méthode permettant de décider la réductibilité inductive dans ce cas.

L'objectif de ce chapitre est de montrer que le problème de réductibilité inductive (avec la nouvelle notion de réduction) s'exprime bien à l'aide de problèmes comportant équations, diséquations, inéquations et "disinéquations" et donc aussi d'ébaucher l'étude de tels problèmes. Une telle étude a déjà été entreprise dans [Ven87] où il est prouvé que les formules générales sont indécidables *lorsque* $\leq$ *est interprété comme l'ordre de sous-terme.* Il est aussi prouvé (avec la même interprétation) que le fragment purement existentiel est quant à lui décidable.

Malheureusement, ces résultats ne s'appliquent pas dans notre cas car nous ne voulons pas interpréter $\leq$ comme l'ordre de sous-terme mais comme un ordre de simplification total sur les termes clos. Nous ne pouvons déduire des résultats de [Ven87] ni la décidabilité du fragment purement existentiel ni l'indécidabilité des formules générales. De toutes façons, ce ne sont pas ces formules qui nous intéressent, mais les *problèmes de complément* qui sont bien des formules équationnelles (générales) mais n'en sont qu'un cas particulier (en fait ces formules ne contiennent même pas le fragment purement existentiel).

Notre objectif n'est pas d'étudier en détails les problèmes généraux mais seulement de montrer à l'aide d'exemples où se situent certaines difficultés. Les questions soulevées ouvrent de nombreuses perspectives de recherche et serviront de conclusion à cette thèse.

8.1 Généralisation des problèmes équationnels : introduction de $\geq$ et de $\not\geq$

Nous n'allons pas donner à nouveau une spécification complète d'une nouvelle définition d'un problème équationnel. Il suffit d'ajouter qu'un système peut désormais aussi être une expression de la forme $t \geq u$ ou $t \not\geq u$ où t et u sont deux termes de même sorte. On écrira aussi $t \leq u$ à la place de $u \geq t$, $t > u$ à la place de $t \geq u \wedge t \neq u$ et $t < u$ à la place de $u > t$.

Pour définir la sémantique de tels problèmes il nous suffit de définir l'interprétation de $\geq$. Étant donnée une F-algèbre $\mathcal{A}$, munie d'une relation d'ordre $\geq_{\mathcal{A}}$, une $\mathcal{A}$-substitution σ *valide* $t \geq u$ si et seulement si $t\sigma \geq_{\mathcal{A}} u\sigma$ et elle valide $t \not\geq u$ ssi on n'a pas $t\sigma \geq_{\mathcal{A}} u\sigma$. La notion de $\mathcal{A}$-solution d'un problème équationnel est ainsi étendue à des problèmes comportant équations, diséquations, inéquations et disinéquations. (Ce que nous appelerons encore problème équationnel dans ce chapitre).

Les résultats de correction et d'adéquation des règles du chapitre 3 ne sont pas modifiés par l'introduction de ces nouveaux symboles (comme dans la section 6.3).

On peut aussi ajouter les règles de transformation qui correspondent à la définition des relations d'ordre (Ces règles pourraient aussi figurer dans la définition de l'algèbre des

problèmes):

$$
\begin{array}{lrcl}
(Ref_1) & t \geq t & \mapsto & \top \\
(Ref_2) & t \not\geq t & \mapsto & \bot \\
(AS_1) & t \geq u \wedge u \geq t & \mapsto & t = u \\
(AS_2) & t \not\geq u \vee u \not\geq t & \mapsto & t \neq u \\
(AS_3) & t \geq u \wedge u \not\geq t & \mapsto & t \geq u \wedge t \neq u \\
(AS_4) & t \not\geq u \vee u \geq t & \mapsto & t \not\geq u \vee t = u \\
(Trans_1) & t \geq u \wedge u \geq v & \mapsto & t \geq u \wedge u \geq v \wedge t \geq v \\
(Trans_2) & t \not\geq u \vee u \not\geq v & \mapsto & t \not\geq u \vee u \not\geq v \vee t \not\geq v \\
\ldots & & &
\end{array}
$$

Nous ne prétendons pas que cette liste est complète (en un sens qui resterait d'ailleurs à préciser : quelle est la nature des formes résolues ?)

8.2 Un exemple d'ordre : description des règles de transformation associées

Les règles de transformation ci-dessus, qui sont adéquates dans tous les modèles ne nous suffisent pas puisque nous n'avons aucun moyen d'éliminer les paramètres des inéquations (et disinéquations); nous n'avons aucun moyen non plus de décomposer les inéquations et les disinéquations.

D'autre part, comme nous l'avons montré, c'est une interprétation particulière dans l'algèbre des termes qui nous intéresse. Nous avons donc choisi ici de décrire une interprétation possible qui permet d'une part de résoudre le problème des décompositions d'inéquations et d'autre part correspond à l'application qui nous intéresse: $\mathcal{A}$ sera l'algèbre des termes $T(F)$ et $\geq_{\mathcal{A}}$ l'ordre récursif sur les chemins avec statut lexicographique (voir [Der87] par exemple) que nous noterons $\geq_{lpo}$ et dont nous rappelons ici la définition:

$\geq_F$ est un ordre sur F appelé *précédence* (qui n'est pas nécessairement total).

$$s \equiv f(s_1, \ldots, s_m) \geq_{lpo} g(t_1, \ldots, t_n) \equiv t \quad \text{ssi}$$

- ou bien $s \equiv t$
- ou bien $\exists i \in \{1, \ldots, m\},\ s_i >_{lpo} t$
- ou bien $f >_F g$ et $\forall j \in \{1, \ldots, n\},\ s >_{lpo} t_j$
- ou bien $f = g$ et $\forall j \in \{1, \ldots, n\},\ s >_{lpo} t_j$ et $(s_1, \ldots, s_m) \gg_{lpo} (t_1, \ldots, t_n)$

où $\gg_{lpo}$ désigne l'extension lexicographique de $\geq_{lpo}$.

$\geq_{lpo}$ est un ordre de simplification (cf [Der87] par exemple). Si de plus $\geq_F$ est un ordre total, on montre facilement (par récurrence sur la profondeur cumulée de s et t) que $\geq_{lpo}$ est total sur $T(F)$. Enfin, cet ordre est stable par substitution i.e.

$$\forall s, t \in T(F, X), \forall \sigma \in \Sigma,\ \ t \geq_{lpo} s \ \Rightarrow\ t\sigma \geq_{lpo} s\sigma$$

On peut alors énoncer les règles de décomposition qui découlent de cette définition :

$$(DI_1)\quad f(s_1,\ldots,s_m) > g(t_1,\ldots,t_n) \;\mapsto\; s_1 \geq g(t_1,\ldots,t_n) \vee \ldots \vee s_m \geq g(t_1,\ldots,t_n) \quad \text{si } g >_F f$$

$$(DI_2)\quad f(s_1,\ldots,s_m) \geq g(t_1,\ldots,t_n) \;\mapsto\; f(s_1,\ldots,s_m) > t_1 \wedge \ldots \wedge f(s_1,\ldots,s_m) > t_n \quad \text{si } f >_F g$$

$$(DI_3)\quad f(s_1,\ldots,s_n) > f(t_1,\ldots,t_n) \;\mapsto\; f(s_1,\ldots,s_n) > t_1 \wedge \ldots \wedge f(s_1,\ldots,s_n) > t_n \wedge (s_1 > t_1 \vee (s_1 = t_1 \wedge s_2 > t_2) \vee \ldots \vee (s_1 = t_1 \wedge \ldots \wedge s_{n-1} = t_{n-1} \wedge s_n > t_n))$$

On peut ajouter à ces règles toutes celles qu'on obtient en remplaçant les $>$ à gauche par des $\geq$ (ou inversement) et en modifiant le membre droit en conséquence.

Bien sûr, de telles règles sont adéquates dans $T(F)$ (ou $T(F,X)$). Si l'on suppose que $\geq_F$ est un ordre total sur F, on peut ne pas considérer les disinéquations puisqu'alors, l'ordre étant total, la règle

$$s \not\geq t \mapsto t \geq s \wedge t \neq s$$

est fortement adéquate dans $T(F)$.

L'interpétation de $\geq$ comme un ordre récursif sur les chemins, total sur les termes fermés, possède ainsi des propriétés remarquables (règles de transformation "locales" esquissées ci-dessus, élimination des disinéquations) que n'a pas, par exemple, l'ordre de sous-termes. Les résultats d'indécidabilité de [Ven87] ne doivent donc pas nous décourager.

On peut aussi envisager une règle semblable à l'explosion qui permet de remplacer une inéquation $s > x$ par un nombre fini d'égalités lorsqu'il n'y a qu'un nombre fini de termes plus petits que s (ce que l'on peut facilement décider à l'aide de $\geq_F$).

Une telle règle, en plus des décompositions, permet de se ramener à des inéquations de la forme $x > t$ ou bien $t > x$, t ayant un nombre infini de majorants et un nombre infini de minorants et x étant une variable. Dans les deux cas, on espère que l'inéquation a une infinité de solutions, ce qui est effectivement le cas si l'on utilise les règles de tests d'occurrence sur les inéquations;

$$(O_3)\quad s > t[s] \;\mapsto\; \bot$$
$$(O_4)\quad s < t[s] \;\mapsto\; \top$$

Ce système de règles permet très probablement d'obtenir des formes résolues ayant au moins une solution fermée lorsque l'on part de problèmes sans paramètre (ce qui permet donc d'obtenir un résultat de décidabilité analogue à celui de [Ven87]).

Il reste néanmoins à étudier le problème de l'élimination des paramètres. Ce qui ne semble pas insurmontable a priori.

8.3 Problèmes de réductibilité inductive : Un exemple

Reprenant la définition de $\rightarrow_{\mathcal{R},E}$ donnée ci-dessus, on peut remarquer que, si $\geq$ est total sur les termes fermés (ce qui est le cas de $\geq_{lpo}$), et si $\rightarrow_{\mathcal{R}} \subseteq >$, alors $\rightarrow_{\mathcal{R},E}$ est noethérien sur les termes fermés.

Nous nous intéressons ici à la réductibilité inductive. De même que dans le chapitre 5, nous définissons la réductibilité inductive par :

s est inductivement réductible par $\mathcal{R}, E$ si, pour toute substitution σ telle que $s\sigma \in T(F)$, il existe un terme t tel que $s \rightarrow_{\mathcal{R},E} t$.

Pour décider ce problème nous allons essayer, comme dans le chapitre 5, de construire une grammaire du langage des termes fermés irréductibles qui sont des instances de s. Et pour construire cette grammaire, nous allons résoudre des problèmes équationnels comportant des inéquations. Etudions un exemple (nous ne développerons pas le cas général):

Exemple 8.1
$F = \{0 : \rightarrow \underline{s};\ \ f : \underline{s} \rightarrow \underline{s};\ \ m : \underline{s} \times \underline{s} \times \underline{s} \rightarrow \underline{s}\}$
$E = \{m(x,y,z) == m(y,z,x)\},$
$\mathcal{R} = \{m(x,x,y) \rightarrow x;\ \ f(f(0)) \rightarrow 0\}$

L'ordre considéré est l'ordre $\geq_{lpo}$ de la section précédente avec la précédence $m >_F f >_F 0$. On peut remarquer que le système $(\mathcal{R}, E)$ n'est pas convergent sur les termes fermés alors que, par complétion sans échec, il est possible d'obtenir un tel système à partir de $\mathcal{R}, E$. Mais cela n'a pas d'importance pour notre problème.

Nous nous intéressons à la réductibilité inductive de $m(x_1, x_2, x_3)$, procédant de la même façon que dans le chapitre 5: nous cherchons les solutions dans NF de:

$$\begin{aligned} \forall y_1, y_2, y_3 : \quad & m(x_1,x_2,x_3) \neq m(y_1,y_1,y_2) \\ & \wedge (m(x_1,x_2,x_3) \neq m(y_1,y_2,y_3) \vee m(y_2,y_3,y_1) \geq m(y_1,y_2,y_3)) \\ & \wedge (m(x_1,x_2,x_3) \neq m(y_1,y_2,y_3) \vee m(y_3,y_1,y_2) \geq m(y_1,y_2,y_3)) \end{aligned}$$

La première diséquation signifie la non réductibilité en tête par la règle $m(x,x,y) \rightarrow x$, les deux disjonctions qui suivent signifient la non-réductibilité en tête par l'équation $m(x,y,z) == m(y,z,x)$, selon qu'elle est consisdérée de gauche à droite ou de droite à gauche.

On peut remarquer que cette formalisation généralise celle du chapitre 5. (On retrouve la précédente lorsqu'il n'y a pas d'équations dans E). On peut aisément définir le problème à résoudre dans le cas général.

Voyons comment transformer ce problème (généralisé):

Par décomposition et élimination des paramètres (EP_2) on obtient le problème:

$$x_1 \neq x_2 \wedge m(x_1,x_2,x_3) \leq m(x_2,x_3,x_1) \wedge m(x_1,x_2,x_3) \leq m(x_3,x_1,x_2)$$

Remarquons qu'il a été possible d'éliminer les paramètres sans difficulté[1].

Par décompositions des inéquations, on obtient ensuite:

$$\begin{aligned} x_1 \neq x_2 \quad & \wedge m(x_2,x_3,x_1) \geq x_1 \wedge m(x_2,x_3,x_1) \geq x_2 \wedge m(x_2,x_3,x_1) \geq x_3 \\ & \wedge m(x_3,x_1,x_2) \geq x_1 \wedge m(x_3,x_2,x_1) \geq x_2 \wedge m(x_3,x_2,x_1) \geq x_3 \\ & \wedge (x_2 > x_1 \vee (x_1 = x_2 \wedge x_3 > x_2) \vee (x_1 = x_2 \wedge x_3 = x_2 \wedge x_1 \geq x_3)) \\ & \wedge (x_3 > x_1 \vee (x_1 = x_3 \wedge x_1 > x_2) \vee (x_1 = x_3 \wedge x_2 = x_1 \wedge x_2 \geq x_3)) \end{aligned}$$

[1]Ceci est, en fait, toujours possible lorsque toutes les équations $s == t$ de E vérifient $Var(s) = Var(t)$. En effet, la simplification (à l'aide des règles du chapitre 3) des diséquations permet de faire apparaitre tous les paramètres comme membre d'une diséquation (et ce, dans chaque disjonction) puis d'appliquer la règle d'élimination des paramètres. Le problème serait plus difficile si l'équation était, par exemple, $m(y_1,y_2,y_3) == m(y_1,y_1,y_3)$. Alors le problème de complément associé contiendrait la disjonction $m(x_1,x_2,x_3) \neq m(y_1,y_1,y_3) \vee m(y_1,y_1,y_3) \leq m(y_1,y_2,y_3)$ qui conduit au problème $x_1 \neq x_2 \vee m(x_1,x_1,x_3) \leq m(x_1,y_2,x_3)$ où le paramètre y_2 n'a pas été éliminé.

En appliquant les règles esquissées dans la section précédente.

Par application des tests d'occurrence le problème ci-dessus se simplifie:

$$\begin{aligned} x_1 \neq x_2 \quad & \wedge(x_2 > x_1 \vee (x_1 = x_2 \wedge x_3 > x_2) \vee (x_1 = x_2 \wedge x_3 = x_2 \wedge x_1 \geq x_3)) \\ & \wedge(x_3 > x_1 \vee (x_1 = x_3 \wedge x_1 > x_2) \vee (x_1 = x_3 \wedge x_2 = x_1 \wedge x_2 \geq x_3)) \end{aligned}$$

Ce problème s'écrit successivement (par mise en forme normale conjonctive et application des règles d'antisymétrie, de réflexivité et de transitivité sur les inéquations):

$$\begin{aligned} x_1 \neq x_2 \quad & \wedge x_2 \geq x_1 \wedge (x_2 > x_1 \vee x_3 \geq x_2) \wedge (x_2 > x_1 \vee x_3 > x_2 \vee x_1 \geq x_3) \\ & \wedge x_3 \geq x_1 \wedge (x_3 > x_1 \vee x_1 \geq x_2) \wedge (x_3 > x_1 \vee x_1 > x_2 \vee x_2 \geq x_3) \end{aligned}$$

Puis

$$x_2 > x_1 \wedge x_3 > x_1$$

Et l'on peut considérer ce dernier problème comme étant en forme résolue.

On en déduit la règle de grammaire conditionnelle:

$$NF_{m(x_1,x_2,x_3)} \rightarrow m(NF_{x_1}, NF_{x_2}, NF_{x_3}) \text{ Si } x_2 > x_1 \text{ et } x_3 > x_1$$

On voit sur cet exemple que la généralisation des résultats des chapitres précédents semble bien s'amorcer. Mais nous avons laissé de nombreuses questions sans réponse. Essayons d'en dresser une liste:

1. Définir les représentants de problèmes équationnels comportant des inéquations et donner les règles permettant de trouver ces représentants (bien entendu ces règles doivent terminer). C'est ce que nous avons commencé à faire en énonçant sous forme de règles de transformation les propriétés d'une relation d'ordre. Mais telles que nous les avons énoncées elles ne terminent pas.

2. Compléter l'énoncé des règles associées à une interprétation de $\geq$ comme ordre de simplification total sur les termes fermés.

3. Trouver un contrôle pour lequel l'application des règles termine et obtenir un résultat de complétude vis à vis des problèmes sans paramètres. L'existence d'un tel algorithme n'est pas assurée. Voyons sur un exemple le problème posé. Soit le problème $\forall y : x \geq f(y)$ (on suppose que F ne contient qu'une constante : 0 et un symbole de fonction unaire : f.) Si l'on essaye d'appliquer la même technique que dans le chapitre 3, l'idée pour éliminer le paramètre est de le "faire remonter" à la racine et d'appliquer une règle d'élimination pour les inéquations $y > t$. Mais ni cette règle d'élimination, ni "faire remonter" le paramètre n'est évident. Si l'on essaye en effet d'appliquer une technique semblable à celles que nous avons déjà employées, nous "explosons" x,. L'un des problèmes obtenus est alors

$$\exists w, \forall y : x = f(w) \wedge f(w) \geq f(y)$$

Malheureusement, la décomposition des inéquations ne permet pas d'obtenir $w > y$. (En cela, on retrouve des problèmes analogues à ceux que nous avons rencontrés dans le chapitre précédent). On obtient en fait:

$$\exists w \forall y : x = f(w) \wedge w \geq f(y) \wedge w \geq y$$

Et ce problème contient le problème dont on est parti. Néanmoins, dans l'exemple présent, il est clair que le problème n'a aucune solution car il est possible de trouver y tel que $x < f(y)$.

4. Il parait possible sans difficulté de généraliser la méthode de calcul des grammaires de formes normales. Mais la décision du vide est une autre affaire puisque les conditions comportent désormais des inéquations. Le vide est il encore décidable dans ce cas ?

Bibliographie

[Aho68] A. V. Aho. Indexed grammars. An extension of context-free grammars. *Journal of the ACM*, 15(4):647–671, October 1968.

[Bac88] L. Bachmair. Proof by consistency in equational theories. In *Proc. 3rd IEEE Symp. Logic in Computer Science, Edinburgh*, July 1988.

[BDH86] L. Bachmair, N. Dershowitz, and J. Hsiang. Orderings for equational proofs. In *Proc. 1st IEEE Symp. Logic in Computer Science, Cambridge, Mass.*, June 1986.

[BE86] D. Bert and R. Echahed. Design and implementation of a generic, logic and functional programming language. In *Proc. ESOP 86, Saarbrücken, LNCS 213*, Springer-Verlag, March 1986. Available as IMAG Research Report 560.

[Ber79] D. Bert. *La Programmation Générique.* Thèse d'Etat, Univ. Grenoble, France, June 1979.

[Ber83] C. Berge. *Graphes.* Gauthier-Villars, third edition, 1983.

[BHS87] H. J. Bürckert, A. Herold, and M. Schmidt-Schauss. On equational theories, unification and decidability. In *Proc. Rewriting Techniques and Applications 87, Bordeaux, LNCS 256*, pages 204–215, Springer-Verlag, May 1987.

[Bir35] G. Birkhoff. On the structure of abstract algebras. In *Proc. Cambridge Phil. Society, 31*, 1935.

[BMPT87] R. Barbuti, P. Mancarella, D. Pedreschi, and F. Turini. Intensional negation of logic programs: examples and implementation techniques. In *Proc. CFLP, Pisa, LNCS 250*, Springer-Verlag, March 1987.

[Bur88] H. J. Bürckert. Solving disequations in equational theories. In *Proc. 9th Conf. on Automated Deduction, Argonne, LNCS 310*, Springer-Verlag, May 1988.

[Caf86] R. Caferra. Notes sur la logique, sa mécanisation, la programmation logique. 1986. Polycopié des cours donnés à l'ENSIMAG, Grenoble.

[CL88] H. Comon and P. Lescanne. *Equational Problems and Disunification.* Research Report 727, LIFIA-IMAG, Grenoble, May 1988. To appear in J. Symbolic Computation.

[Cla78] K. L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, Plenum, New York, 1978.

[Col82] A. Colmerauer. *Prolog II. Manuel de référence et modèle théorique.* Research Report, GIA Luminy, Marseille, March 1982.

[Col84] A. Colmerauer. Equations and inequations on finite and infinite trees. In *FGCS'84 Proceedings*, pages 85–99, November 1984.

[Com86] H. Comon. Sufficient completeness, term rewriting systems and anti-unification. In *Proc. 8th Conf. on Automated Deduction, Oxford, LNCS 230*, pages 128–140, Springer-Verlag, July 1986.

[Com88] H. Comon. Inductive proofs by specifications transformation. 1988. To appear in Proc. RTA 89.

[Cou81] B. Courcelle. Fundamental properties of infinite trees. In *Proc. Int. Summer School on Theoretical Foundations of Programming Methodology, Munich*, 1981.

[Dau84] M. Dauchet. *Cours d'informatique théorique.* Université de Lille, 1984.

[Der87] N. Dershowitz. Termination of rewriting. *J. Symbolic Computation*, 3(1):69–115, February 1987.

[DJ88] N. Dershowitz and J.-P. Jouannaud. Term rewriting systems. In *Handbook of Theoretical Computer Science*, Pitman, 1988. To appear.

[DM79] N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8):465–476, August 1979.

[EKP78] H. Ehrig, H. J. Kreowski, and P. Padawitz. Stepwise specifications and implementations of abstract data types. In *Proc. 5th ICALP, LNCS 62*, pages 205–226, 1978.

[ES77] J. Engelfriet and E.M. Schmidt. IO and OI. *Journal of Computer and System Sciences*, 15, 1977.

[Fag87] F. Fages. Associative-commutative unification. *J. Symbolic Computation*, 3(3), June 1987.

[FGJM85] K. Futatsugi, J. Goguen, J.-P. Jouannaud, and J. Meseguer. Principles of OBJ2. In *Proc. 12th ACM Symp. Principles of Programming Languages, New Orleans*, 1985.

[Fin86] A. Finkel. *Structuration des Systèmes de Transitions. Applications au Contrôle du Parallelisme par Files FIFO.* Thèse de Doctorat, Univ. Orsay, France, 1986.

[Fri86] L. Fribourg. A strong restriction of the inductive completion procedure. In *Proc. 13th ICALP, Rennes, LNCS 226*, pages 105–115, Springer-Verlag, 1986.

[Gal86] J. H. Gallier. *Logic for Computer Science: Foundations of Automatic Theorem Proving.* Harper and Row, 1986.

[GB85] J. H. Gallier and R. V. Book. Reductions in tree replacement systems. *Theoretical Computer Science*, 37:123–150, 1985.

[GH78] J. V. Guttag and J. J. Horning. The algebraic specification of abstract data types. *Acta Informatica*, 10:27–52, 1978.

[GHM78] Guttag, Horowitz, and Musser. Abstract data types and software validation. *Communications of the ACM*, 21(12), December 1978.

[GKK88] I. Gnaedig, H. Kirchner, and C. Kirchner. Equational completion in order-sorted algebras. In *Proc. CAAP 88, Nancy, LNCS 299*, Springer-Verlag, March 1988.

[GM87a] J. Goguen and J. Meseguer. *Order-Sorted Algebra I: Partial and Overloaded Operators, Errors and Inheritance.* Draft, Computer Science Lab., SRI International, 1987.

[GM87b] J. Goguen and J. Meseguer. Order-sorted algebra solves the constructor-selector, multiple representation and coercion problems. In *Proc. 2nd IEEE Symp. Logic in Computer Science, Ithaca, NY*, June 1987.

[Gog80] J. A. Goguen. How to prove inductive hypothesis without induction. In *Proc. 5th Conf. on Automated Deduction, Les Arcs, LNCS 87*, July 1980.

[GS87] J. Gallier and W. Snyder. A general complete E-unification procedure. In *Proc. Rewriting Techniques and Applications 87, Bordeaux, LNCS 256*, Springer-Verlag, May 1987.

[GTW78] J. A. Goguen, J. W. Thatcher, and E. G. Wagner. An initial algebra approach to the specification, correctness and implementation of abstract data types. In *Current Trends in Programming Methodology, vol. 4*, pages 80–149, Prentice Hall Int., 1978.

[Her30] J. Herbrand. *Recherches sur la théorie de la démonstration.* Thèse d'Etat, Univ. Paris, 1930. Also in: *Ecrits logiques de Jacques Herbrand*, PUF, Paris, 1968.

[HH82] G. Huet and J.-M. Hullot. Proofs by induction in equational theories with constructors. *Journal of Computer and System Sciences*, 25(2), 1982.

[HO80] G. Huet and D. Oppen. Equations and rewrite rules: a survey. In R. Book, editor, *Formal Language Theory: Perspectives and Open Problems*, pages 349–405, Academic Press, 1980.

[HR87] H. Hsiang and M. Rusinowitch. On word problems in equational theories. In *Proc. 14th ICALP, Karlsruhe, LNCS 267*, Springer-Verlag, 1987. Available as INRIA Research Report 678.

[Hue76] G. Huet. *Résolution d'équations dans les langages d'ordre* $1, 2, \ldots \omega$. Thèse d'Etat, Univ. Paris 7, 1976.

[Hue81] G. Huet. A complete proof of correctness of the Knuth-Bendix completion algorithm. *Journal of Computer and System Sciences*, 23:11–21, 1981.

[JK86a] J.-P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal on Computing*, 15(4), 1986.

[JK86b] J.-P. Jouannaud and E. Kounalis. Automatic proofs by induction in equational theories without constructors. In *Proc. 1st IEEE Symp. Logic in Computer Science, Cambridge, Mass.*, June 1986.

[JL87] J. Jaffar and J.-L. Lassez. Constraint logic programming. In *Proc. 14th ACM Symp. Principles of Programming Languages, Munich*, 1987.

[Jor86] Ph. Jorrand. Term rewriting as a basis for the design of a functional and parallel programming language. A case study: the language FP2. In *Fundamentals of Artificial Intelligence, LNCS 232*, pages 221–276, Springer-Verlag, 1986.

[KB70] D. E. Knuth and P. B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297, Pergamon Press, 1970.

[Kir85] C. Kirchner. *Méthodes et Outils de Conception Systématique d'Algorithmes d'Unification dans les Théories equationnelles.* Thèse d'Etat, Univ. Nancy, France, 1985.

[Kir86] C. Kirchner. Computing unification algorithms. In *Proc. 1st IEEE Symp. Logic in Computer Science, Cambridge, Mass.*, pages 206–216, 1986.

[Kir88] C. Kirchner. Order-sorted equational unification. In *Proc. Int. Conference on Logic Programming*, 1988.

[KK88] C. Kirchner and H. Kirchner. Constraint equational reasoning. 1988. Submitted.

[KKM88] C. Kirchner, H. Kirchner, and J. Meseguer. *Operational Semantics of OBJ-3.* Technical Report, CRIN, 1988.

[KL87] C. Kirchner and P. Lescanne. Solving disequations. In *Proc. 2nd IEEE Symp. Logic in Computer Science, Ithaca, NY*, pages 347–352, 1987.

[KNZ85] D. Kapur, P. Narendran, and H. Zhang. *On Sufficient Completeness and Related Properties of Term Rewriting Systems.* Research Report, General Electric Company, October 1985. Preprint.

[KNZ86] D. Kapur, P. Narendran, and H. Zhang. Complexity of sufficient completeness. 1986. To appear in TCS.

[Kou85] E. Kounalis. Completeness in data type specifications. In *Proc. EUROCAL 85, Linz, LNCS 204*, pages 348–362, Springer-Verlag, April 1985.

[Kuc88] G. A. Kucherov. A new quasi-reducibility testing algorithm and its applications to proofs by induction. In *Proc. 1st Workshop on Algebraic and Logic Programming, Gaussig*, November 1988.

[Kun87] K. Kunen. *Signed Data Dependencies in Logic Programs.* Tech. Report 719, Univ. Wisconsin, Madison, October 1987.

[Llo84] J. W. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, 1984.

[LLT86] A. Lazrek, P. Lescanne, and J.-J. Thiel. *Proving Inductive Equalities. Algorithms and Implementation.* Research Report, CRIN, Nancy, France, 1986. To appear in Information and Computation.

[LM87] J.-L. Lassez and K. G. Marriott. Explicit representation of terms defined by counter examples. *J. Automated Reasoning*, 3(3):1–17, September 1987.

[LMM86] J.-L. Lassez, M. J. Maher, and K. G. Marriot. Unification revisited. In *Proc. Workshop on Found. of Logic and Functional Programming, Trento, LNCS 306*, Springer-Verlag, December 1986.

[Lug88] D. Lugiez. A deduction procedure for first order programs. 1988. Submitted.

[Mah88a] M. J. Maher. Complete axiomatizations of the algebras of finite, rational and infinite trees. In *Proc. 3rd IEEE Symp. Logic in Computer Science, Edinburgh*, pages 348–357, July 1988.

[Mah88b] M. J. Maher. Complexity of disunification. 1988. Private communication.

[Mal71] A. I. Mal'cev. Axiomatizable classes of locally free algebras of various types. In *The Metamathematics of Algebraic Systems. Collected Papers. 1936-1967*, pages 262–289, North-Holland, 1971.

[MG85] J. Meseguer and J. Goguen. Initiality, induction and computability. In M. Nivat and J. Reynolds, editors, *Algebraic Methods in Semantics*, chapter 14, Cambridge Univ. Press, 1985.

[Mil78] R. Milner. A theory of type polymorphismn programming. *Journal of Computer and System Sciences*, 17, 1978.

[MM82] A. Martelli and U. Montanari. An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems*, 4(2):258–282, 1982.

[Mus80] D. Musser. Proving inductive properties of abstract data types. In *Proc. 7th ACM Symp. Principles of Programming Languages, Las Vegas*, 1980.

[Pey87] S. Peyton-Jones. *The Implementation of Functional Programming Languages.* Prentice Hall Int., 1987.

[Pla85] D. Plaisted. Semantic confluence tests and completion methods. *Information and Control*, 65:182–215, 1985.

[PW78] M. S. Paterson and M. N. Wegman. Linear unification. *Journal of Computer and System Sciences*, 16, 1978.

[Rei78] R. Reiter. On closed world data bases. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, Plenum, New York, 1978.

[Sch86] M. Schmidt-Schauss. Unification in many-sorted equational theory. In *Proc. 8th Conf. on Automated Deduction, Oxford, LNCS 230*, pages 538–552, Springer-Verlag, July 1986.

[Sch87a] M. Schmidt-Schauss. Unification in an order-sorted calculus with declarations. 1987. Lecture presented at Workshop on Unification, Val d'Ajoul, France.

[Sch87b] Ph. Schnoebelen. Rewriting techniques for the temporal analysis of communicating processes. In *Proc. PARLE 87, vol. II: Parallel Languages, Eindhoven, LNCS 259*, pages 402–419, Springer-Verlag, June 1987.

[Sch88a] Ph. Schnoebelen. Refined compilation of pattern-matching for functional languages. *Science of Computer Programming*, 11(2):133–159, December 1988.

[Sch88b] Ph. Schnoebelen. Refined compilation of pattern-matching for functional languages. In *Proc. 1st Workshop on Algebraic and Logic Programming, Gaussig*, November 1988.

[Sie84] J. Siekmann. Universal unification. In *Proc. 7th Conf. on Automated Deduction, Napa, LNCS 87*, Springer-Verlag, May 1984.

[SNGM87] G. Smolka, W. Nutt, J. Goguen, and J. Meseguer. *Order-Sorted Equational Computation.* SEKI Report SR-87-14, Univ. Kaiserslautern, December 1987.

[Sti81] M. Stickel. A unification algorithm for associative-commutative functions. *Journal of the ACM*, 28(3):423–434, 1981.

[Sti86] M. Stickel. An introduction to automated deduction. In *Fundamentals of Artificial Intelligence, LNCS 232*, Springer-Verlag, 1986.

[Sto77] Joseph E. Stoy. *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory.* MIT Press, 1977.

[Thi84] J.-J. Thiel. Stop loosing sleep over incomplete specifications. In *Proc. 11th ACM Symp. Principles of Programming Languages, Salt Lake City*, 1984.

[Uni87] *Proc. Workshop on Unification, Val d'Ajol*, 1987.

[Uni88] *Proc. Worshop on Unification, Val d'Ajol*, 1988.

[Ven87] K. N. Venkataraman. Decidability of the purely existential fragment of the theory of term algebras. *Journal of the ACM*, 34(2):492–510, 1987.

[Wal85] C. Walther. A mecanical solution of Schubert's streamroller by many-sorted resolution. *Artificial Intelligence*, 26(2):217–224, 1985.

Index

INPG département des études doctorales

A U T O R I S A T I O N de S O U T E N A N C E

VU les dispositions de l'article 15 Titre III de l'arrêté du 5 juillet 1984 relatif aux études doctorales

VU les rapports de présentation de Messieurs
. J. GALLIER, Professeur
. J.P JOUANNAUD, Professeur

Monsieur COMON Hubert

est autorisé(e) à présenter une thèse en soutenance en vue de l'obtention du diplôme de DOCTEUR de L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE, spécialité "Informatique"

Fait à Grenoble, le 7 mars 1988

Georges LESPINARD
Président
de l'Institut National Polytechnique
de Grenoble

P.O. le Vice-Président,

Auteur: Hubert COMON

Etablissement: Laboratoire d'Informatique Fondamentale et d'Intelligence Artificielle (LIFIA)

Titre : Unification et disunification. Théorie et applications.

Résumé : Une *formule équationnelle* est une formule du premier ordre dont le seul symbole de prédicat est l'égalité. Nous donnons un ensemble de règles de transformation de telles formules et étudions leur correction dans divers modèles. Nous étudions ensuite plusieurs contrôles sur ces règles qui permettent d'établir des résultats de terminaison et de complétude vis à vis d'ensembles de *formes résolues*. Plusieurs notions de formes résolues sont envisagées.

Une conséquence de ces résultats est, par exemple, la décidabilité de la validité dans l'univers de Herbrand d'une formule équationnelle quelconque. En d'autres termes, nous proposons une axiomatisation complète des arbres finis sur un alphabet fini. De plus les résultats de terminaison sont établis pour un contrôle "minimal" et plusieurs algorithmes de *disunification* peuvent être obtenus par raffinement de ce contrôle.

Les résultats précédents sont étendus aux arbres rationnels, aux algèbres avec sortes ordonnées et à certaines théories équationnelles.

Nous nous intéressons à plusieurs applications. La principale d'entre elles étant l'étude du problème de la correction des spécifications algébriques et plus précisément la décision de la réductibilité inductive. Nous montrons comment, à l'aide la simplification de certaines formules équationnelles, il est possible de calculer une grammaire (conditionnelle) du langage des termes fermés irréductibles pour un système de réécriture. Nous proposons ensuite un algorithme de décision du vide pour de telles grammaires, obtenant ainsi un algorithme de décision de la réductibilité inductive dans le cas général.

Mots clés: Unification, démonstration automatique, preuves par induction, spécification algébriques, logique équationnelle.